Migrating Endgame to Elastic Security

A step-by-step guide to successfully migrating from Endgame to Elastic while maintaining endpoint coverage and environmental stability

For SMP Version 3.30

Contents

Introduction

The purpose of this guide is to serve as a pragmatic manual for migrating a currently functioning Endgame sensor management platform and its associated sensor's configurations and settings to Elastic. If the steps below are followed it can be expected that coverage on endpoints will be maintained throughout the process with the final result being a fully functioning Elastic Security cluster utilizing Fleet and Agent with the Defend (endpoint) integration to detect and respond against threats on endpoints previously covered by Endgame.

The features and functionality of Endgame and Elastic are different from one another, so there is not always a 1-to-1 direct mapping from a feature and its configuration in Endgame to Elastic. Regardless of differences in the implementation details, Elastic Security + Defend offers the same amount of protection with its modernized approach to detection, response actions, and monitoring capabilities.

High Level Overview

The breakdown of steps needed to start and complete the migration can be summarized as follows:

1. Perform a sizing exercise to ensure there are the appropriate number of Fleet servers for the desired number of Endpoints and deploy the needed Fleet Servers.
2. If Endgame is currently running in an air-gapped environment, implement a self-hosted package repository and artifact registry
3. Configure Index Lifecycle Management to age data appropriately through hot/warm/cold/frozen data tiers
4. Configure a Snapshot repository
5. Configure Snapshot Lifecycle Management to age cluster snapshots appropriately
6. Review and update trusted applications in Endgame in preparation for import into Elastic Security
7. Export and Import trusted applications and host isolations exceptions from Endgame in to Elastic Security using the API
8. Export and import the exception list and block list from Endgame in Elastic Security using the API
9. Create new trusted applications in Elastic Security to avoid conflicts with running Endgame processes
10. Create new trusted applications in Endgame to avoid conflicts with running Elastic Defend and Agent processes
11. Create “detect only” (no prevention) agent policy with Endpoint Security + Osquery
12. Identify hosts currently running Endgame that will run Elastic Endpoint security at the same time

13. Identify a set of test users across teams that can confirm their applications work as expected
14. Deploy agents with "detect only" (no prevention) policy to a group of test endpoints that are representative of a standard host environment
15. Using agent metrics monitoring, take note of baseline agent CPU and memory consumption
16. Soak period:

- Have UAT users confirm everything functions as expected
- Triage alerts, add exceptions, compare to alerts generated in Endgame
- Verify CPU and memory consumption are in line with expectations

17. Repeat steps 14, 15, and 16 on a new group of endpoints until all endpoints are covered
18. Once verified to be stable, alter agent policies from "detect only" to "prevent" in groups that have had an adequate soak period
19. Uninstall Endgame

Migration Process

Each section of this guide contains detailed steps and information for every step of the migration process

Add a Fleet Server

To use Fleet for central management, a Fleet Server[1] must be running and accessible to your hosts. There are a few approaches you can take when deploying Fleet Server:

- Provision Fleet Server on Elastic Cloud as part of the hosted Elasticsearch Service. Elastic manages both Fleet Server and Elasticsearch.
- Deploy Fleet Server on-premises to work with Elasticsearch running on-premises. You self-manage both Fleet Server and Elasticsearch.
- Deploy Fleet Server on-premises to work with a hosted Elasticsearch Service. You manage Fleet Server and Elastic manages Elasticsearch.

Read more about each to choose the best approach for your situation.

Provision Fleet Server on Elastic Cloud

Fleet Server can be provisioned and hosted on Elastic Cloud. In this case, when the deployment is created, a highly available set of Fleet Servers is automatically deployed.

This approach might be right for you if you want to reduce on-prem compute resources and you'd like Elastic to take care of provisioning and life cycle management of your deployment.

[1] https://www.elastic.co/guide/en/fleet/current/fleet-server.html

With this approach, multiple Fleet Servers are automatically provisioned to satisfy the chosen instance size (instance sizes are modified to satisfy the scale requirement). You can also choose the resources allocated to each Fleet Server and whether you want each Fleet Server to be deployed in multiple availability zones. If you choose multiple availability zones to address your fault-tolerance requirements, those instances are also utilized to balance the load.

This approach might not be right for you if you have restrictions on connectivity to the internet.

For step-by-step instructions, go to Deploy on Elastic Cloud[2].

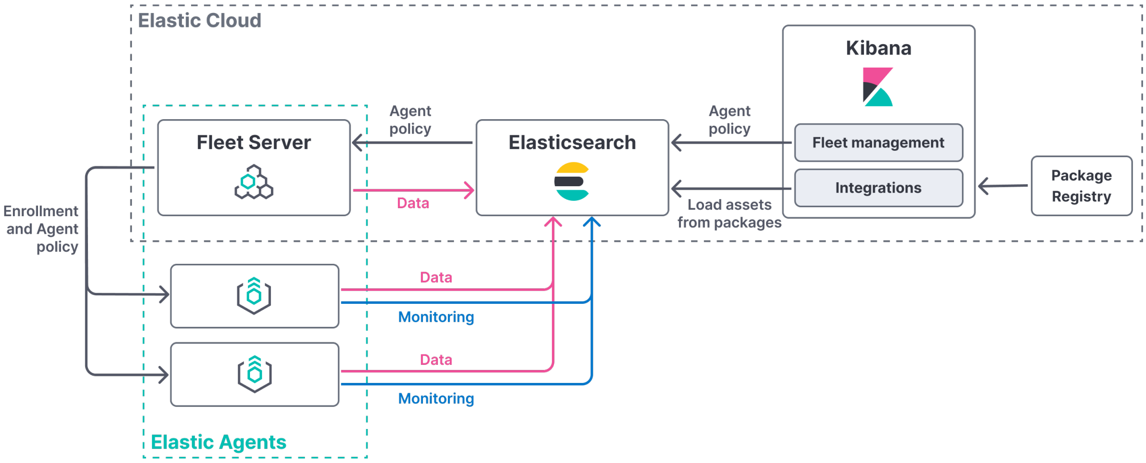


Figure 1: *fleet-server-cloud-deployment.png*

Deploy Fleet Server on-premises

Alternatively, you can deploy Fleet Server on-premises and manage it yourself. In this deployment model, you are responsible for high-availability, fault-tolerance, and lifecycle management of Fleet Server.

This approach might be right for you if you would like to limit the control plane traffic out of your data center or have requirements for fully air-gapped operations. For example, you might take this approach if you need to satisfy data governance requirements or you want agents to only have access to a private segmented network.

This approach might not be right for you if you don't want to manage the life cycle of your Elastic environment and instead would like that to be handled by Elastic.

When using this approach, it's recommended that you provision multiple instances of the Fleet Server and use a load balancer to better scale the deployment. You also have the option to use your organization's certificate to establish a secure connection from Fleet Server to Elasticsearch.

For step-by-step instructions, go to Deploy on-premises and self-managed[3].

[2] https://www.elastic.co/guide/en/fleet/current/add-fleet-server-cloud.html
[3] https://www.elastic.co/guide/en/fleet/current/add-fleet-server-on-prem.html

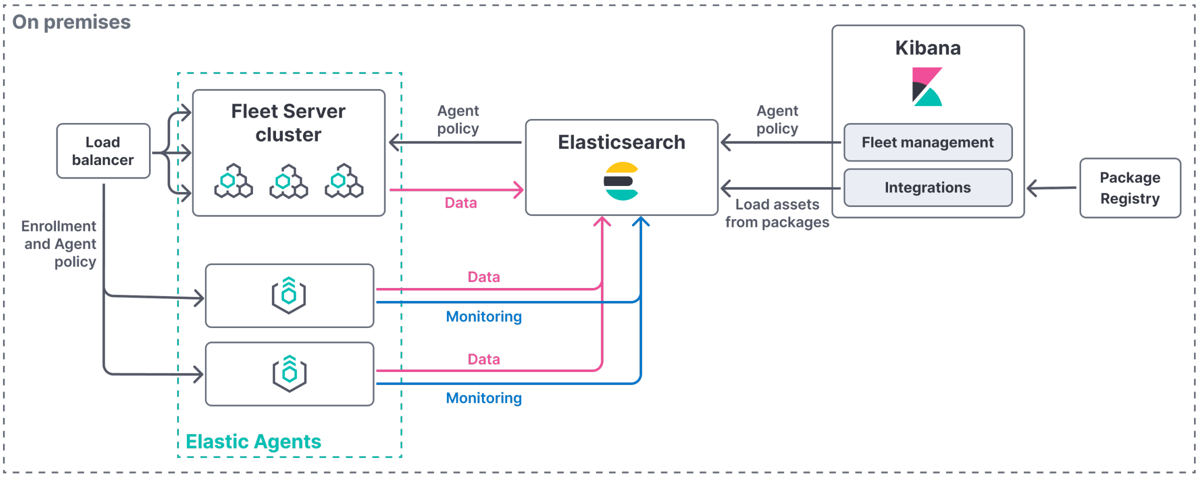


Figure 2: *fleet-server-on-prem-deployment.png*

Deploy Fleet Server on-premises to work with a hosted Elasticsearch Service

Another approach is to deploy a cluster of Fleet Servers on-premises and connect them back to Elastic Cloud with access to Elasticsearch and Kibana. In this deployment model, you are responsible for high-availability, fault-tolerance, and lifecycle management of Fleet Server.

This approach might be right for you if you would like to limit the control plane traffic out of your data center. For example, you might take this approach if you are a managed service provider or a larger enterprise that segregates its networks.

This approach might not be right for you if you don't want to manage the life cycle of an extra compute resource in your environment for Fleet Server to reside on.

For step-by-step instructions, go to Deploy Fleet Server on-premises and Elasticsearch on Cloud[4].

[4] https://www.elastic.co/guide/en/fleet/current/add-fleet-server-mixed.html

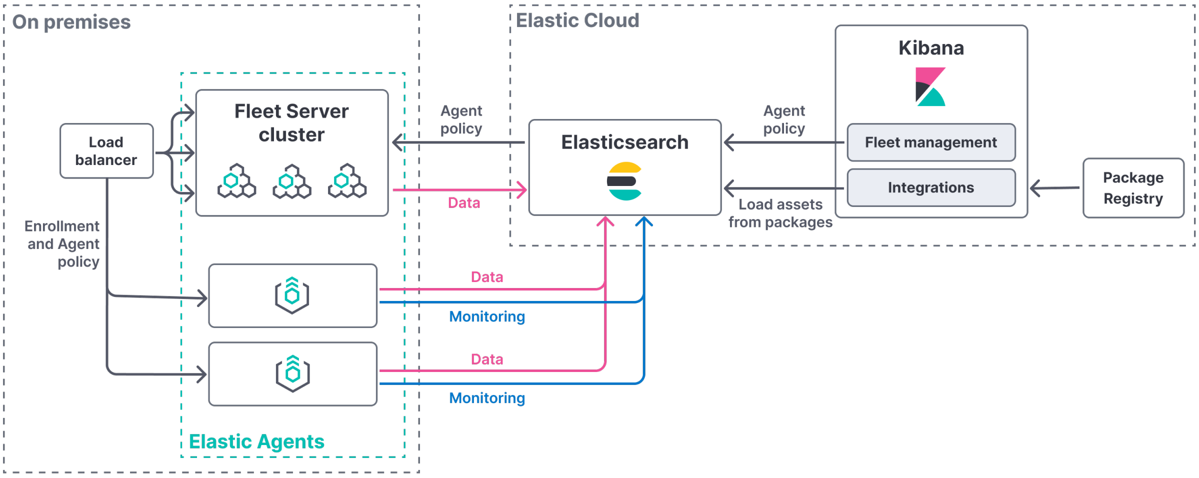


Figure 3: *fleet-server-on-prem-es-cloud.png*

Scaling Fleet

Please see the most up-to-date version of our scaling documentation[5] for current recommendations on scaling Fleet Server to meet the resource requirements needed for the number of Agents that are anticipated to be deployed.

Air-gapped environments

If Elastic will be running in an air-gapped environment, a self-hosted artifact repository and package registry will be needed. The steps in this section will serve as a guide for setting up and configuring the needed pieces to make this possible. Note the examples below are version and operating system specific, so commands may need to be modified to match the version of Elastic and the operating system that Elastic is running on.

When running Elastic Agents in a restricted or closed network, you need to take extra steps to make sure:

- Kibana is able to reach the Elastic Package Registry to download package metadata and content.
- Elastic Agents are able to download binaries during upgrades.

Use a proxy server to access the Elastic Package Registry

By default Kibana downloads package metadata and content from the public Elastic Package Registry at epr.elastic.co[6].

If you can route traffic to the public endpoint of the Elastic Package Registry through a network gateway, set the following property in Kibana to use a proxy server:

[5] https://www.elastic.co/guide/en/fleet/current/fleet-server-scalability.html
[6] https://epr.elastic.co/

```
xpack.fleet.registryProxyUrl: your-nat-gateway.corp.net
```

For more information, refer to Use a proxy server with Elastic Agent and Fleet[7].

Host your own Elastic Package Registry

> Note: The Elastic Package Registry packages include signatures used in package verification[8]. By default, Fleet uses the Elastic public GPG key to verify package signatures. If you ever need to change this GPG key, use the `xpack.fleet.packageVerification.gpgKeyPath` setting in `kibana.yml`. For more information, refer to Fleet settings[9].

If routing traffic through a proxy server is not an option, you can host your own Elastic Package Registry.

The Elastic Package Registry can be deployed and hosted onsite using one of the available Docker images. These docker images include the Elastic Package Registry and a selection of packages.

There are different distributions available. **The examples below reference Elastic version 8.8.1. Please make the necessary changes to choose the version best for your environment.**:

- 8.8.1 (recommended): docker.elastic.co/package-registry/distribution:8.8.1 - Selection of packages from the production repository released with Elastic Stack 8.8.1.
- lite-8.8.1: docker.elastic.co/package-registry/distribution:lite-8.8.1 - Subset of the most commonly used packages from the production repository released with Elastic Stack 8.8.1. This image is a good candidate to start using Fleet in air-gapped environments.
- production: docker.elastic.co/package-registry/distribution:production - Packages available in the production registry (https://epr.elastic.co/[10]).
- lite: docker.elastic.co/package-registry/distribution:lite - Subset of the most commonly used packages available in the production registry (https://epr.elastic.co/[11]).

To update the distribution image, re-pull the image and then restart the docker container.

Every distribution contains packages that can be used by different versions of the Elastic Stack. The Elastic Package Registry API exposes a Kibana version constraint that allows for filtering packages that are compatible with a particular version.

> Note: These steps use the standard Docker CLI, but you can create a Kubernetes manifest based on this information. These images can also be used with other container runtimes compatible with Docker images.

1. Pull the Docker image from the public Docker registry:

```
docker pull docker.elastic.co/package-registry/distribution:8.8.1
```

2. Save the Docker image locally:

[7] https://www.elastic.co/guide/en/fleet/master/fleet-agent-proxy-support.html
[8] https://www.elastic.co/guide/en/fleet/master/package-signatures.html
[9] https://www.elastic.co/guide/en/kibana/master/fleet-settings-kb.html
[10] https://epr.elastic.co
[11] https://epr.elastic.co

```
docker save -o package-registry-8.8.1.tar docker.elastic.co/package-registry/distribution:8.8.1
```

TIP: Check the image size to ensure that you have enough disk space.

3. Transfer the image to the air-gapped environment and load it:

```
docker load -i package-registry-8.8.1.tar
```

4. Run the Elastic Package Registry:

```
docker run -it -p 8080:8080 docker.elastic.co/package-registry/distribution:8.8.1
```

5. (Optional) You can monitor the health of your Elastic Package Registry with requests to the root path:

```
docker run -it -p 8080:8080 \
    --health-cmd "curl -f -L http://127.0.0.1:8080/" \
    docker.elastic.co/package-registry/distribution:8.8.1
```

Connect Kibana to your hosted Elastic Package Registry

Use the `xpack.fleet.registryUrl` property in the Kibana config to set the URL of your hosted package registry. For example:

```
xpack.fleet.registryUrl: "http://package-registry.corp.net:8080"
```

TLS configuration of the Elastic Package Registry

You can configure the Elastic Package Registry to listen on a secure HTTPS port using TLS.

For example, given a key and a certificate pair available in /etc/ssl, you can start the Elastic Package Registry listening on the 443 port using the following command:

```
docker run -it -p 443:443 \
  -v /etc/ssl/package-registry.key:/etc/ssl/package-registry.key:ro \
  -v /etc/ssl/package-registry.crt:/etc/ssl/package-registry.crt:ro \
  -e EPR_ADDRESS=0.0.0.0:443 \
  -e EPR_TLS_KEY=/etc/ssl/package-registry.key \
  -e EPR_TLS_CERT=/etc/ssl/package-registry.crt \
  docker.elastic.co/package-registry/distribution:8.8.1
```

Using custom CA certificates

If you are using self-signed certificates or certificates issued by a custom Certificate Authority (CA), you need to set the file path to your CA in the NODE_EXTRA_CA_CERTS environment variable in the Kibana startup files.

```
NODE_EXTRA_CA_CERTS="/etc/kibana/certs/ca-cert.pem"
```

Host your own artifact registry for binary downloads

Elastic Agents must be able to access the Elastic artifact registry to download binaries during upgrades. By default Elastic Agents download artifacts from https://artifacts.elastic.co/downloads/.

To make binaries available in an air-gapped environment, you can host your own custom artifact registry, and then configure Elastic Agents to download binaries from it.

1. Create a custom artifact registry in a location accessible to your Elastic Agents:

a. Download the latest release artifacts from the public Elastic artifact registry at `https://artifacts.elastic.co/downloads/`. For example, the following cURL commands download all the artifacts that may be needed to upgrade Elastic Agents running on Linux. The exact list depends on which integrations you're using.

```
curl -O https://artifacts.elastic.co/downloads/apm-server/apm-server-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/apm-server/apm-server-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/apm-server/apm-server-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/auditbeat/auditbeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/auditbeat/auditbeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/auditbeat/auditbeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/osquerybeat/osquerybeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/osquerybeat/osquerybeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/osquerybeat/osquerybeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/beats/packetbeat/packetbeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/beats/packetbeat/packetbeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/beats/packetbeat/packetbeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/cloudbeat/cloudbeat-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/cloudbeat/cloudbeat-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/cloudbeat/cloudbeat-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/endpoint-dev/endpoint-security-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/endpoint-dev/endpoint-security-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/endpoint-dev/endpoint-security-8.8.1-linux-x86_64.tar.gz.asc
curl -O https://artifacts.elastic.co/downloads/fleet-server/fleet-server-8.8.1-linux-x86_64.tar.gz
curl -O https://artifacts.elastic.co/downloads/fleet-server/fleet-server-8.8.1-linux-x86_64.tar.gz.sha512
curl -O https://artifacts.elastic.co/downloads/fleet-server/fleet-server-8.8.1-linux-x86_64.tar.gz.asc
```

b. On your HTTP file server, group the artifacts into directories and subdirectories that follow the same convention used by the Elastic artifact registry:

```
<source_uri>/<artifact_type>/<artifact_name>-<version>-<arch>-<package_type>
```

Where `<artifact_type>` may be `beats/elastic-agent`, `beats/filebeat`, `fleet-server`, `endpoint-dev`, and so on.

> TIP: Make sure you have a plan or automation in place to update your artifact registry when new versions of Elastic Agent are available.

2. Add the agent binary download location to Fleet settings:

a. In Kibana, go to **Fleet** → **Settings**.
b. Under **Agent Binary Download**, click **Add agent binary source** to add the location of your artifact registry. For more detail about these settings, refer to Agent binary download settings. If you want all Elastic Agents to download binaries from this location, set it as the default.

3. If your artifact registry is not the default, edit your agent policies to override the default:

a. Go to **Fleet** → **Agent** policies and click the policy name to edit it.
b. Click **Settings**.
c. Under **Agent Binary Download**, select your artifact registry.

When you trigger an upgrade for any Elastic Agents enrolled in the policy, the binaries are downloaded from your artifact registry instead of the public repository.

Create and Apply Index Lifecycle Policy for Endpoint Data Streams

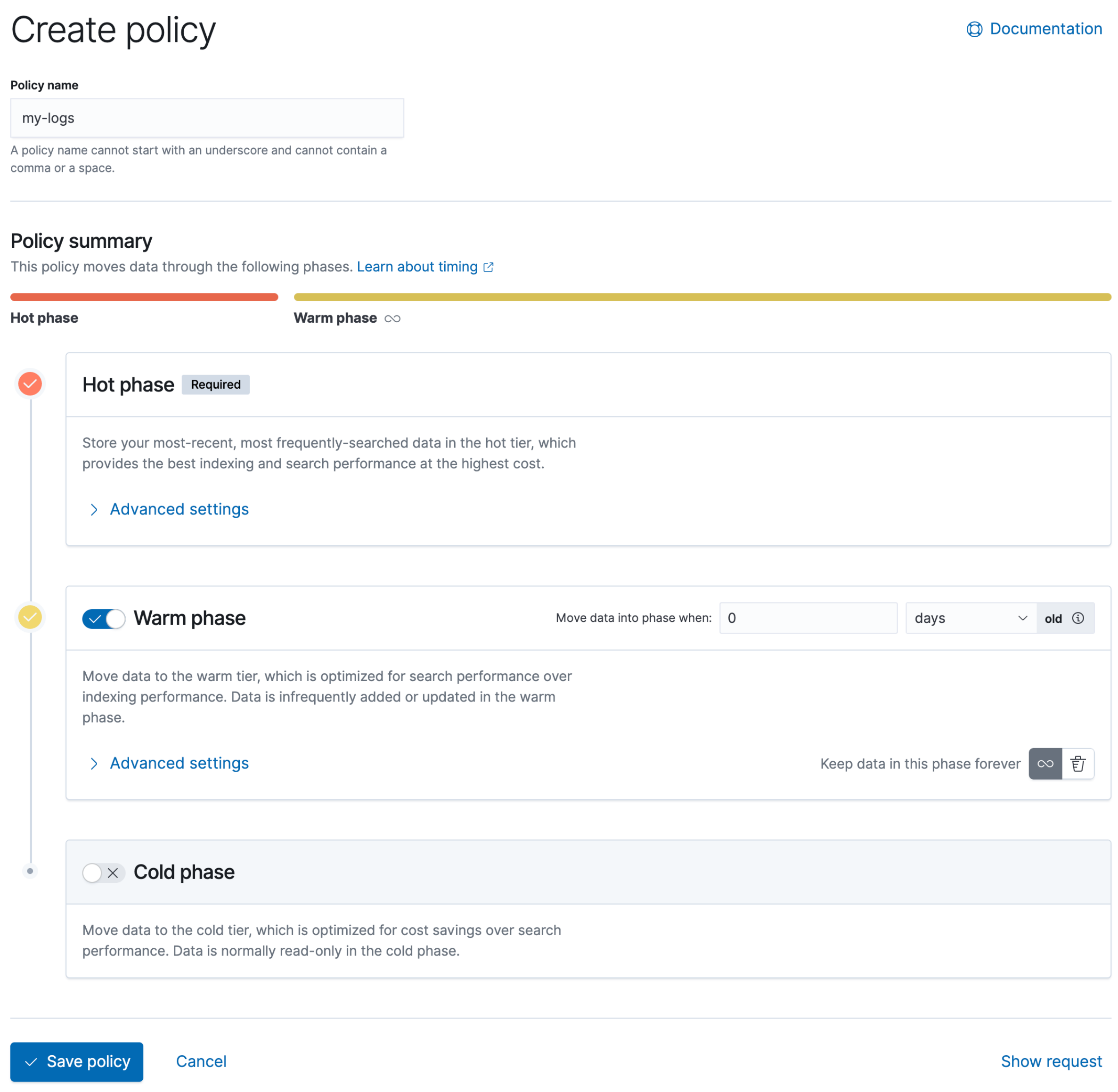
We will start by creating an ILM policy in Elastic with Kibana to age data out of the cluster according to need and requirements. For ILM to manage an index, a valid policy must be specified in the `index.lifecycle.name` index setting. The best way to do this once the policy has been defined is to create a component template[12] that contains the `index.lifecycle.name` index setting, and then clone and modify the existing index template by adding our component template.

[12] https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-component-template.html

ILM policies are stored in the global cluster state and can be included in snapshots by setting `include_global_state` to **`true`** when you take the snapshot[13]. When the snapshot is restored, all of the policies in the global state are restored and any local policies with the same names are overwritten.

Step 1: Create lifecycle policy

To create a lifecycle policy from Kibana, open the menu and go to **Stack Management** > **Index Lifecycle Policies**. Click **Create policy**.



Create policy

Policy name: my-logs

A policy name cannot start with an underscore and cannot contain a comma or a space.

Policy summary

This policy moves data through the following phases. Learn about timing

Hot phase Required

Store your most-recent, most frequently-searched data in the hot tier, which provides the best indexing and search performance at the highest cost.

Warm phase

Move data into phase when: 0 days old

Move data to the warm tier, which is optimized for search performance over indexing performance. Data is infrequently added or updated in the warm phase.

Keep data in this phase forever

Cold phase

Move data to the cold tier, which is optimized for cost savings over search performance. Data is normally read-only in the cold phase.

Figure 4: *ilm_policy.png*

[13] https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-take-snapshot.html

You specify the lifecycle phases for the policy and the actions to perform in each phase.

The create or update policy[14] API is invoked to add the policy to the Elasticsearch cluster.

API example:

```
PUT _ilm/policy/my_policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_primary_shard_size": "25GB"
          }
        }
      },
      "delete": {
        "min_age": "30d",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

1. Roll over the index when it reaches 25 GB in size
2. Delete the index 30 days after rollover

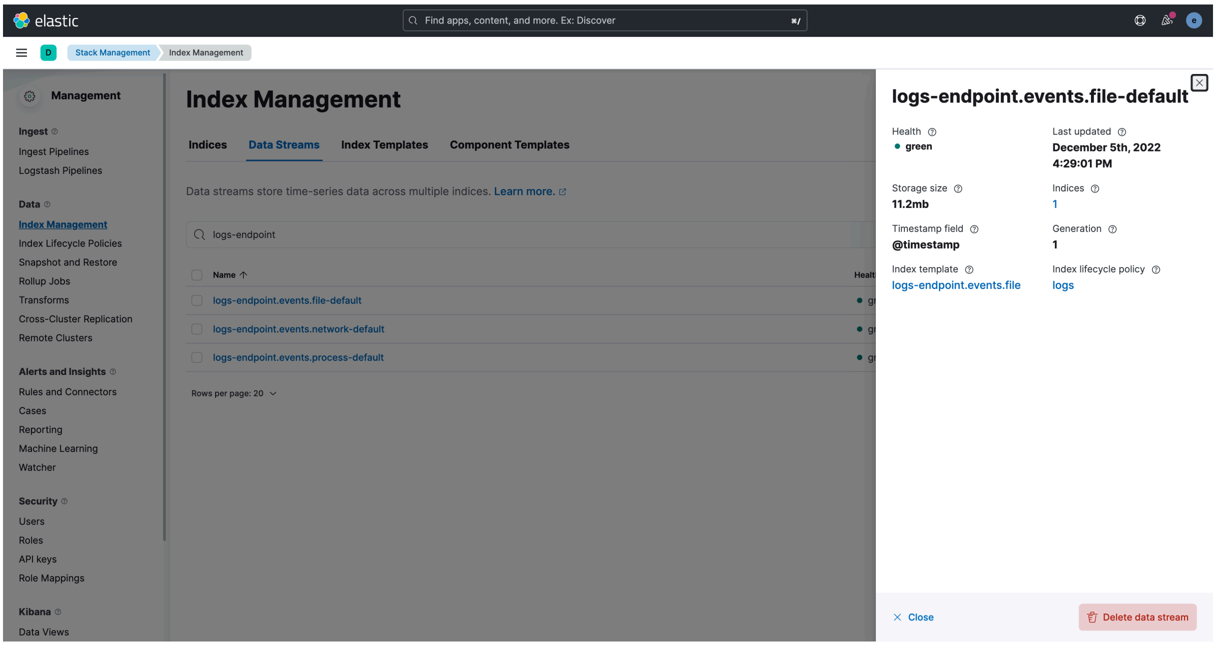
Step 2: View data streams

The **Data Streams** view in Kibana shows you the data streams, index templates, and ILM policies associated with a given integration.

Navigate to **Stack Management** > **Index Management** > **Data Streams**. Search for `logs-endpoint` to see all data streams associated with the Endpoint integration. Select the `logs-endpoint.events.file-{namespace}` data stream to view its associated index template and ILM policy. As you can see, the data stream follows the Data stream naming scheme[15] and starts with its type, `logs-endpoint-`.

[14] https://www.elastic.co/guide/en/elasticsearch/reference/current/ilm-put-lifecycle.html

[15] https://www.elastic.co/guide/en/fleet/current/data-streams.html#data-streams-naming-scheme



Management

Ingest
Ingest Pipelines
Logstash Pipelines

Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Cross-Cluster Replication
Remote Clusters

Alerts and Insights
Rules and Connectors
Cases
Reporting
Machine Learning
Watcher

Security
Users
Roles
API keys
Role Mappings

Kibana
Data Views

Index Management

Indices | Data Streams | Index Templates | Component Templates

Data streams store time-series data across multiple indices. Learn more.

logs-endpoint

Name ↑	Healt
logs-endpoint.events.file-default	● gr
logs-endpoint.events.network-default	● gr
logs-endpoint.events.process-default	● gr

Rows per page: 20

logs-endpoint.events.file-default

Health ● green	Last updated December 5th, 2022 4:29:01 PM
Storage size 11.2mb	Indices 1
Timestamp field @timestamp	Generation 1
Index template logs-endpoint.events.file	Index lifecycle policy logs

Figure 5: *Data Streams*

Step 3: Create a component template

For your changes to continue to be applied in future versions, you must put all custom index settings into a component template. The component template must follow the data stream naming scheme, and end with `@custom`:

```
<type>-<dataset>-<namespace>@custom
```

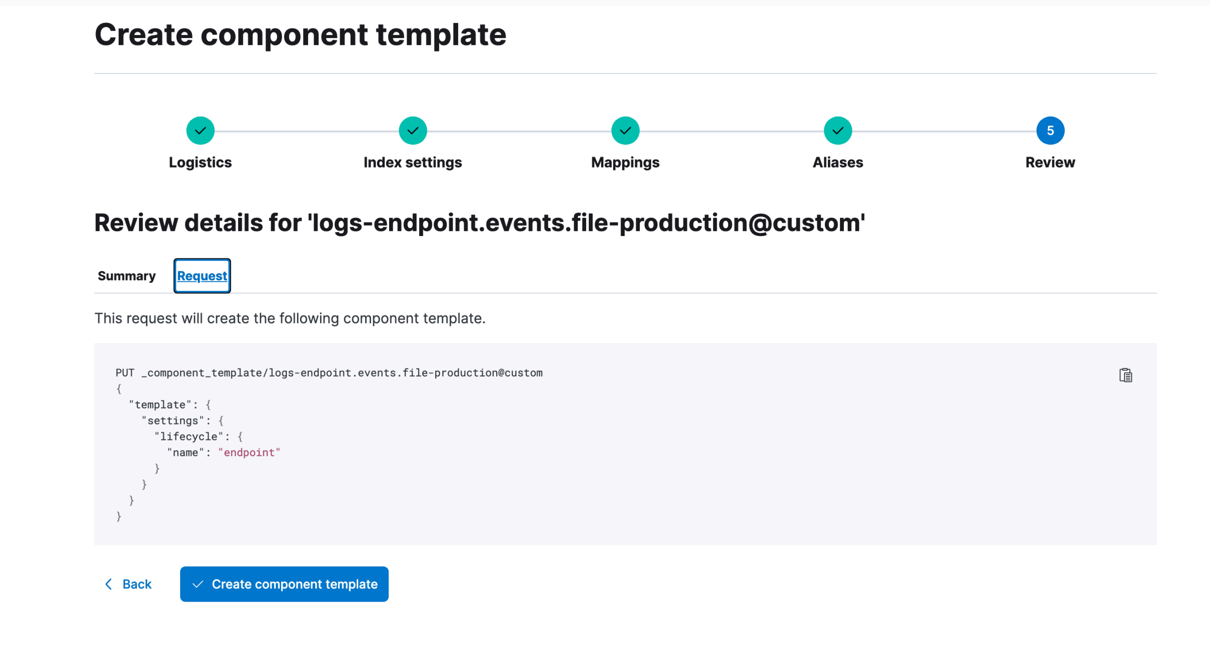
For example, to create custom index settings for the `logs-endpoint.events` data stream with a namespace of production, the component template name would be:

```
logs-endpoint.events.file-production@custom
```

Navigate to **Stack Management** > **Index Management** > **Component Templates** Click **Create component template**. Use the template above to set the name—in this case, `logs-endpoint.events.file-production@custom`. Click **Next**. Under Index settings, set the ILM policy name under the `lifecycle.name` key:

```
{
  "lifecycle": {
    "name": "endpoint"
  }
}
```

Continue to **Review** and ensure your request looks similar to the image below. If it does, click **Create component template**.



Review details for 'logs-endpoint.events.file-production@custom'

This request will create the following component template.

```
PUT _component_template/logs-endpoint.events.file-production@custom
{
  "template": {
    "settings": {
      "lifecycle": {
        "name": "endpoint"
      }
    }
  }
}
```

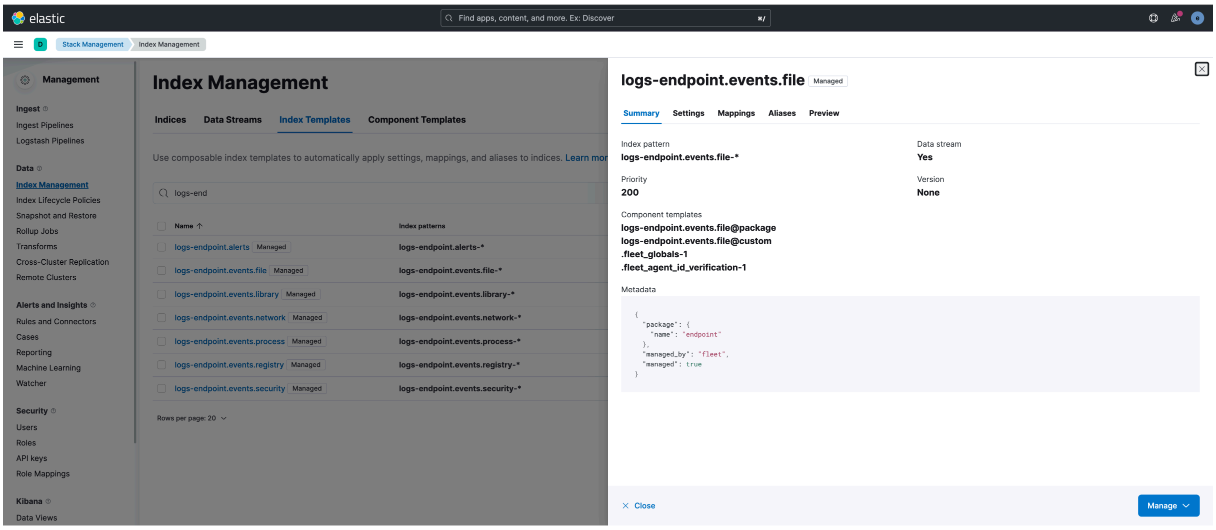
Figure 6: *Component Template*

Step 4: Clone and modify the existing index template

Now that you've created a component template, you need to create an index template to apply the changes to the correct data stream. The easiest way to do this is to duplicate and modify the integration's existing index template.

> WARNING: When duplicating the index template, do not change or remove any managed properties. This may result in problems when upgrading.

1. Navigate to **Stack Management** > **Index Management** > **Index Templates**.
2. Find the index template you want to clone. The index template will have the <`type`> and <`dataset`> in its name, but not the <`namespace`>. In this case, it's `logs-endpoint.events.file`.
3. Select **Actions** > **Clone**.
4. Set the name of the new index template to `logs-endpoint.events.file-production`.
5. Change the index pattern to include a namespace—in this case, logs-endpoint.events.file-production*. This ensures the previously created component template is only applied to the `production` namespace.
6. Set the priority to `250`. This ensures that the new index template takes precedence over other index templates that match the index pattern.
7. Under Component templates, search for and add the component template created in the previous step. To ensure your namespace-specific settings are applied over other custom settings, the new template should be added below the existing `@custom` template.
8. Create the index template.



Index Management

Use composable index templates to automatically apply settings, mappings, and aliases to indices. Learn mor

logs-end

Name	Index patterns
logs-endpoint.alerts Managed	logs-endpoint.alerts-*
logs-endpoint.events.file Managed	logs-endpoint.events.file-*
logs-endpoint.events.library Managed	logs-endpoint.events.library-*
logs-endpoint.events.network Managed	logs-endpoint.events.network-*
logs-endpoint.events.process Managed	logs-endpoint.events.process-*
logs-endpoint.events.registry Managed	logs-endpoint.events.registry-*
logs-endpoint.events.security Managed	logs-endpoint.events.security-*

Rows per page: 20

logs-endpoint.events.file Managed

Index pattern: logs-endpoint.events.file-*

Data stream: Yes

Priority: 200

Version: None

Component templates:
logs-endpoint.events.file@package
logs-endpoint.events.file@custom
.fleet_globals-1
.fleet_agent_id_verification-1

Metadata

```
{
  "package": {
    "name": "endpoint"
  },
  "managed_by": "fleet",
  "managed": true
}
```

Figure 7: *Index Template*

Register a Snapshot Repository

Most security use cases require endpoint data to be kept for long periods of time. Snapshots allow for keeping data in low cost storage long after the data has aged out of the cluster with ILM or deletion, and can be restored into the cluster at any time if needed. You must register a repository before you can take or restore snapshots.

Please refer to the most relevant and up-to-date documentation[16] on registering a snapshot repository to ensure the desired data-retention and backup policies have been applied.

Note: If you are using Elastic Cloud then a snapshot repository and policy are automatically created and this step is not necessary.

Review and Update Trusted Applications in Endgame

This step is to ensure trusted applications in Endgame have been reviewed and updated in preparation for import into Elasticsearch.

Please review each trusted application: 1. Verify the format is correct for each entry 2. Delete any trusted application entries that are incorrectly formatted or that are no longer needed.

---

[16] https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-register-repository.html

Migrate Policies Trusted Applications, Host Isolation Exceptions, Blocklists, and the Exception List from Endgame SMP to Elastic Security directly

As of version 3.30 of the Endgame Sensor Management Platform administrators can now migrate Policies, Trusted Applications, Host Isolation Exceptions, Blocklists, and Exceptionlist to Kibana directly using the `migrate` API. This is different from the `export` API. While the migrate API endpoint has been around in prior releases the difference is that the Exception List is now available to be migrated and it is included in the `all` endpoint.

A user can now use a POST request to the migrate API (example: https://endgame-smp.com/api/v1/migrate/blocklists) with a body that contains a Kibana URL, username, and password

Example:

```
{
    "url": "<uri>",
    "user": "<user>",
    "password": "<password>"
}
```

Migrate API Endpoints include:

```
/api/v1/migrate/trusted-apps/
/api/v1/migrate/host-isolation-exceptions/
/api/v1/migrate/blocklists/
/api/v1/migrate/exceptionlist/
/api/v1/migrate/policies/
/api/v1/migrate/all/
/api/v1/migrate/status/
/api/v1/migrate/status/job_id
```

Status page for migrate jobs There is new status endpoint to track state of started migrate jobs. Each call for migrate will return now a list of associated job_ids. A customer can query for all or for specific job using following GET API

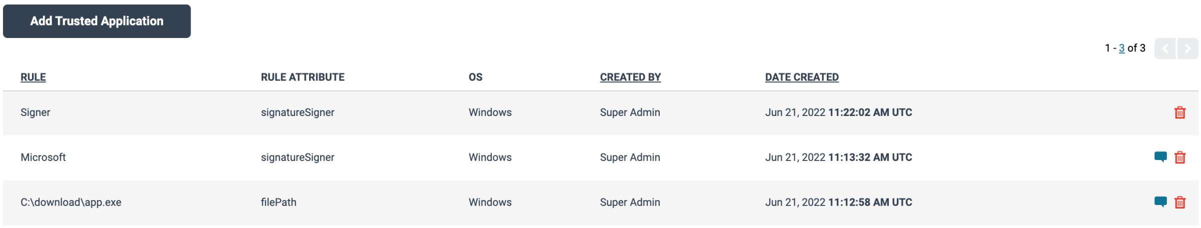
```
/api/v1/migrate/status/
/api/v1/migrate/status/job_id
```

Response from status API call shows all possible states for a given job

```
{
    "code": 200,
    "result": {
        "total_running": 0,
        "jobs": {
            "finished": {
                "4ede12bb-0151-4001-8ede-f18f8ab4536d": {
                    "skipped": 0,
                    "started": "2023-05-23T12:29:18.37222422Z",
                    "finished": "2023-05-23T12:29:30.812307693Z",
                    "failed_transform": 0,
                    "failed_sent": 0,
                    "total": 3,
                    "type": "policies",
                    "sent": 3
                },
            },
            "running": {}
        },
        "total_jobs": 1
    }
}
```

Duplicates during migration There was an update on migrate functionality to not allow duplicated entries in Kibana anymore. The check happens on internal Kibana field, which is updated using migrate API starting from 3.30 version, therefore if you have migrated some data using 3.2x SMP version, it will not be de-duped.

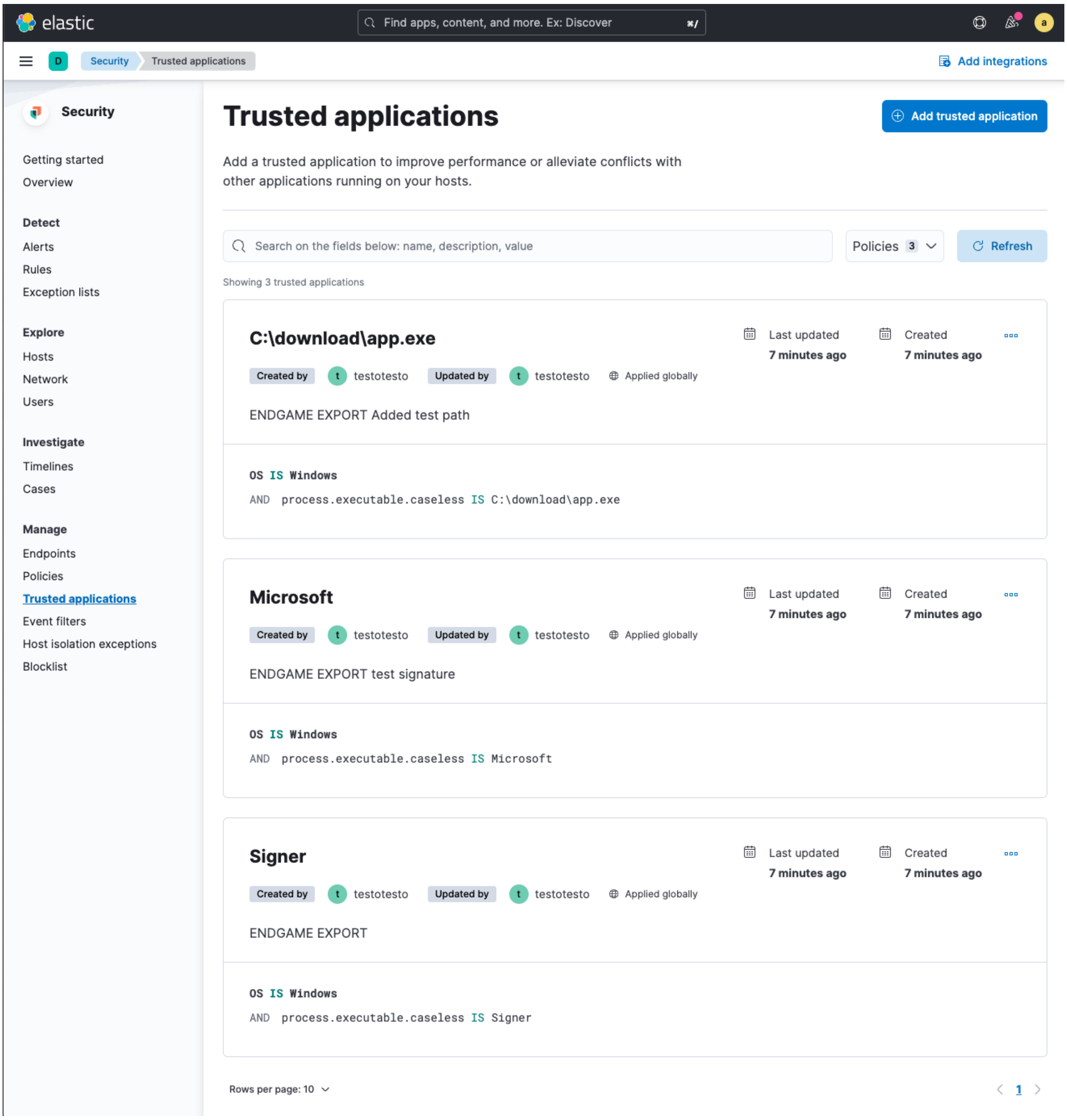
Once the migration has been performed, please compare the views both visually: `https://<smphost>:<smpport>/admin/trusted-applications`



RULE	RULE ATTRIBUTE	OS	CREATED BY	DATE CREATED
Signer	signatureSigner	Windows	Super Admin	Jun 21, 2022 **11:22:02 AM UTC**
Microsoft	signatureSigner	Windows	Super Admin	Jun 21, 2022 **11:13:32 AM UTC**
C:\download\app.exe	filePath	Windows	Super Admin	Jun 21, 2022 **11:12:58 AM UTC**

Figure 8: *Endgame Trusted Applications*

`https://<kibanahost>:<kibanaport>/app/security/administration/trusted_apps`



Trusted applications

Add a trusted application to improve performance or alleviate conflicts with other applications running on your hosts.

Showing 3 trusted applications

C:\download\app.exe

Created by testotesto Updated by testotesto Applied globally

Last updated 7 minutes ago Created 7 minutes ago

ENDGAME EXPORT Added test path

```
OS IS Windows
AND  process.executable.caseless IS C:\download\app.exe
```

Microsoft

Created by testotesto Updated by testotesto Applied globally

Last updated 7 minutes ago Created 7 minutes ago

ENDGAME EXPORT test signature

```
OS IS Windows
AND  process.executable.caseless IS Microsoft
```

Signer

Created by testotesto Updated by testotesto Applied globally

Last updated 7 minutes ago Created 7 minutes ago

ENDGAME EXPORT

```
OS IS Windows
AND  process.executable.caseless IS Signer
```

Figure 9: *Elastic Trusted Applications*

—

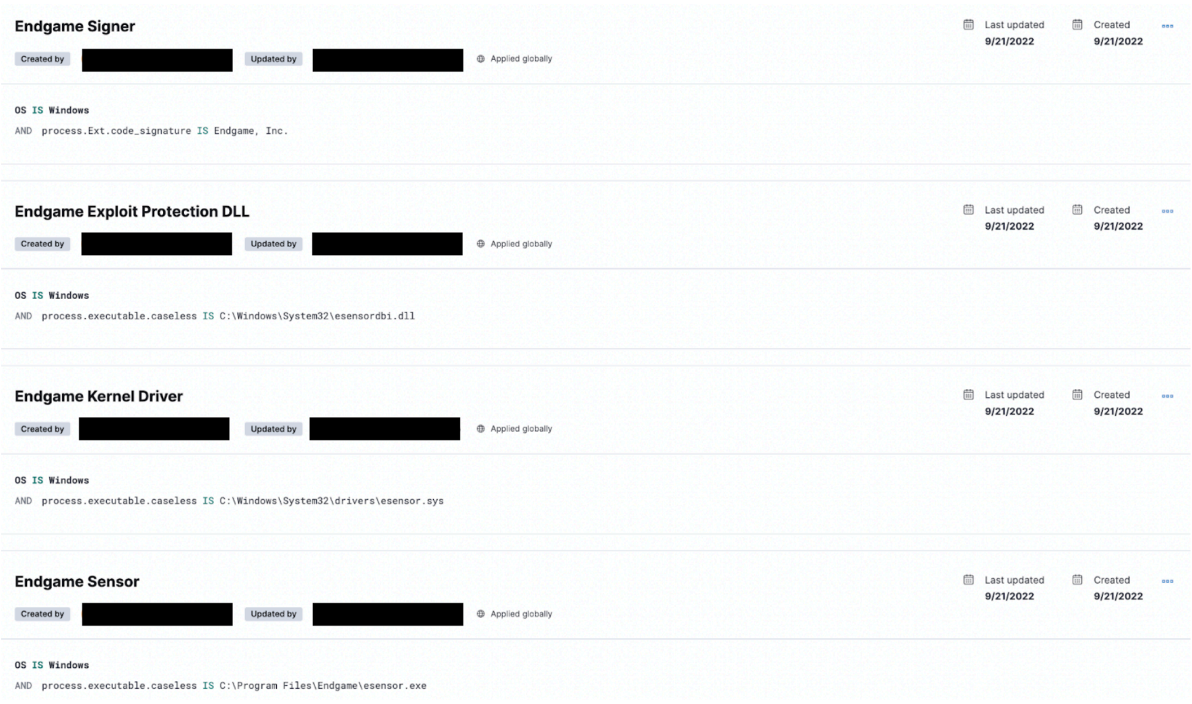
Export and Import Trusted Paths into Elastic

NOTE: This product feature is not yet available in Elastic. This section will be updated once Endgame Policy Trusted Paths can be exported in a format that is then able to be imported and used in Elastic Security.

Create New Trusted Applications in Elastic for Endgame Processes

Elastic Defend will generate alerts for the Endgame sensor if both are running on the same host. To prevent conflicts or issues Trusted Application entries should be created for Endgame process executables within Elastic.

Here is an example of Trusted Application entries in Elastic for the Windows version of the Endgame Sensor:



Endgame Signer

Created by [redacted] Updated by [redacted] Applied globally

Last updated 9/21/2022 Created 9/21/2022

```
OS IS Windows
AND process.Ext.code_signature IS Endgame, Inc.
```

Endgame Exploit Protection DLL

Created by [redacted] Updated by [redacted] Applied globally

Last updated 9/21/2022 Created 9/21/2022

```
OS IS Windows
AND process.executable.caseless IS C:\Windows\System32\esensordbi.dll
```

Endgame Kernel Driver

Created by [redacted] Updated by [redacted] Applied globally

Last updated 9/21/2022 Created 9/21/2022

```
OS IS Windows
AND process.executable.caseless IS C:\Windows\System32\drivers\esensor.sys
```

Endgame Sensor

Created by [redacted] Updated by [redacted] Applied globally

Last updated 9/21/2022 Created 9/21/2022

```
OS IS Windows
AND process.executable.caseless IS C:\Program Files\Endgame\esensor.exe
```

Figure 10: *Elastic Trusted Application for Endgame*

For Windows the following entries will need to be added as trusted applications:

- Signiture IS `Endgame, Inc.`
- Path IS `C:\Windows\System32\esensordbi.dll`
- Path IS `C:\Windows\System32\drivers\esensor.sys`
- Path IS `C:\Program Files\Endgame\esensor.exe`

For Linux the following entries will need to be added as trusted applications:

- Path IS `/usr/sbin/esensor`
- Path IS `/var/opt/esensor`

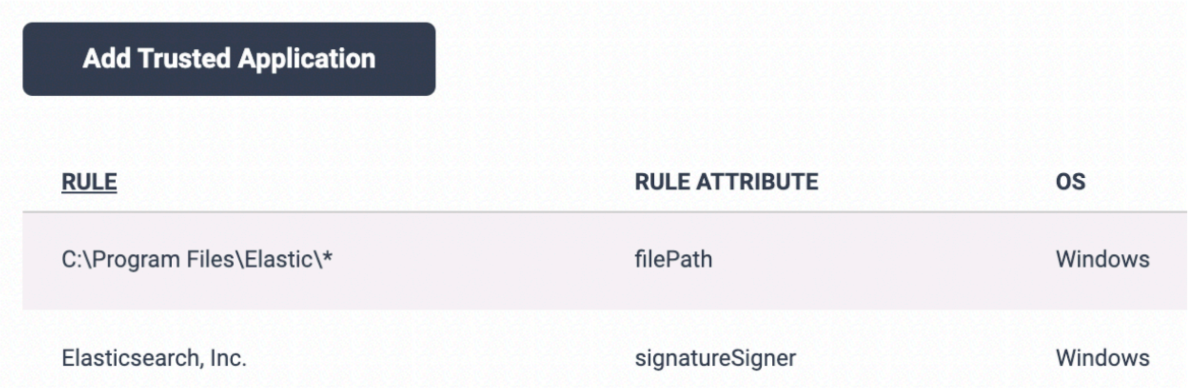
For Mac the following entries will need to be added as trusted applications:

- Path IS `/Library/Endgame/esensor`
- Path IS `/Library/Endgame`

Create New Trusted Applications in Endgame for Agent/Defend Processes

Similar to the step above, Endgame will also generate alerts for Elastic Defend if both are running on the same host. To prevent conflicts or issues Trusted Application entries should be created for Elastic executables within Endgame.

Here is an example of Trusted Application entries in Endgame for the Windows version of the Elastic:



RULE	RULE ATTRIBUTE	OS
C:\Program Files\Elastic\*	filePath	Windows
Elasticsearch, Inc.	signatureSigner	Windows

Figure 11: *Endgame Trusted Application for Elastic*

For Windows the following entries will need to be added as trusted applications:

- Signiture IS `Elasticsearch, Inc.`
- Path IS `C:\Program Files\Elastic\*`

For Mac the following entries will need to be added as trusted applications:

- Path IS `/Library/Elastic/Agent/*`
- Path IS `/usr/bin/elastic-agent`

Endgame does not support trusted applications on Linux

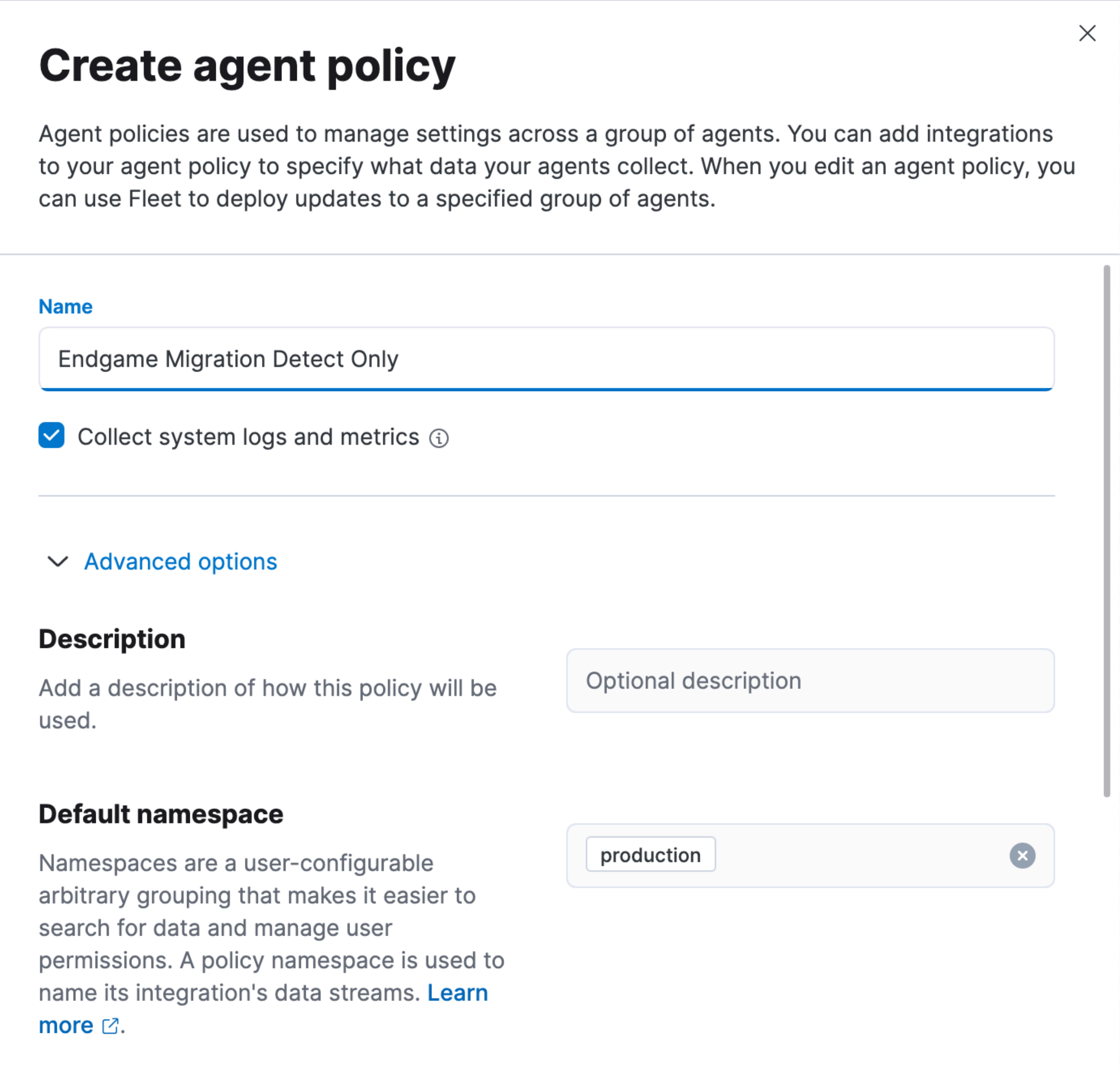
Create “Detect Only” Agent Policy with Endpoint Security + Osquery

The goal of this step is to create an Elastic Agent policy by adding the Defend (Endpoint and Cloud Security) integration and the Osquery Manager integration. Once the defend integration has been added we need to ensure whatever protections are enabled have their protection level set to `Detect` so that Agent does not interfere with Endgame policies that have response actions while we are testing.

Step 1: Create Agent Policy

In Kibana, open the menu on the left-hand side and navigate to **Fleet** > **Agent policies** and select the **Create agent policy** button. Give it a name such as **“Endgame Migration Detect Only”** and open **Advanced**

options to assign it a namespace that aligns with the ILM policy that was created earlier. Once complete, click the **Create agent policy** button at the bottom of the page to create the policy.



Create agent policy

Agent policies are used to manage settings across a group of agents. You can add integrations to your agent policy to specify what data your agents collect. When you edit an agent policy, you can use Fleet to deploy updates to a specified group of agents.

Name: Endgame Migration Detect Only

☑ Collect system logs and metrics

Advanced options

Description

Add a description of how this policy will be used.

Optional description

Default namespace

Namespaces are a user-configurable arbitrary grouping that makes it easier to search for data and manage user permissions. A policy namespace is used to name its integration's data streams. Learn more.

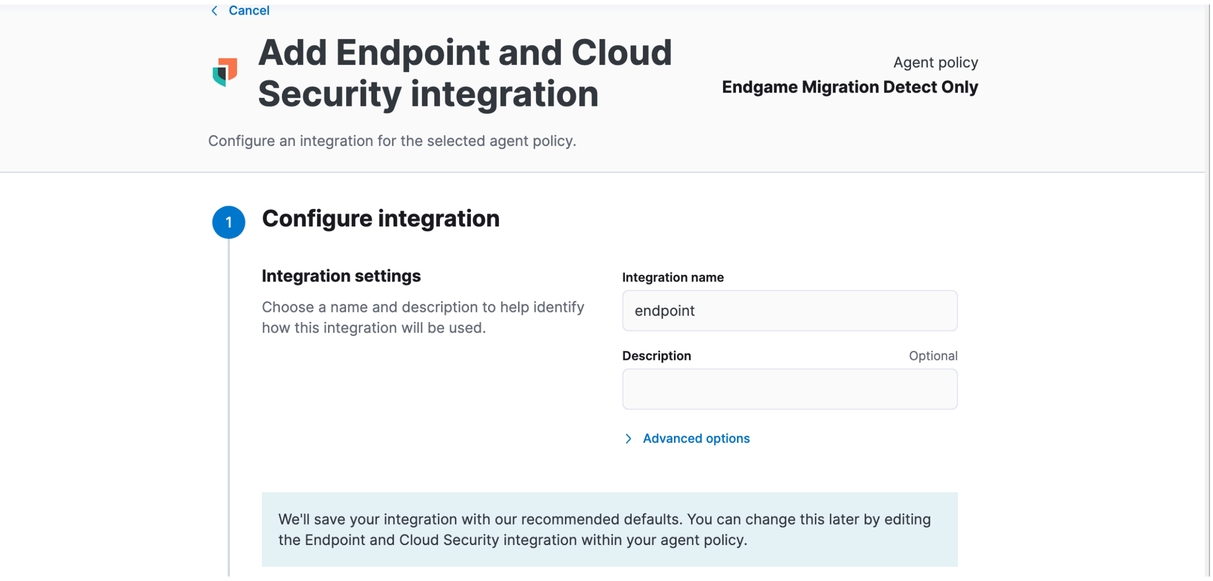
production

Figure 12: *Detect Only Policy*

Step 2: Add the Defend (Endpoint and Cloud Security) Integration

1. On the Agent policies page, click on the name of the agent policy that was just created.
2. Now click on **Add integration** on the right-hand side of the page. Once on the integrations page, click on the **Endpoint and Cloud Security** card pinned to the top-right side of the page.
3. At this point you should be on the Endpoint and Cloud Security integration overview page. Click on the blue **Add Endpoint and Cloud Security** button on the top-right of the page.

4. Give the integration a name that will help you identify the integration like "endpoint" and click **Save and continue** at the bottom-right of the page.



< Cancel

Add Endpoint and Cloud Security integration

Agent policy
Endgame Migration Detect Only

Configure an integration for the selected agent policy.

1 Configure integration

Integration settings

Choose a name and description to help identify how this integration will be used.

Integration name: endpoint

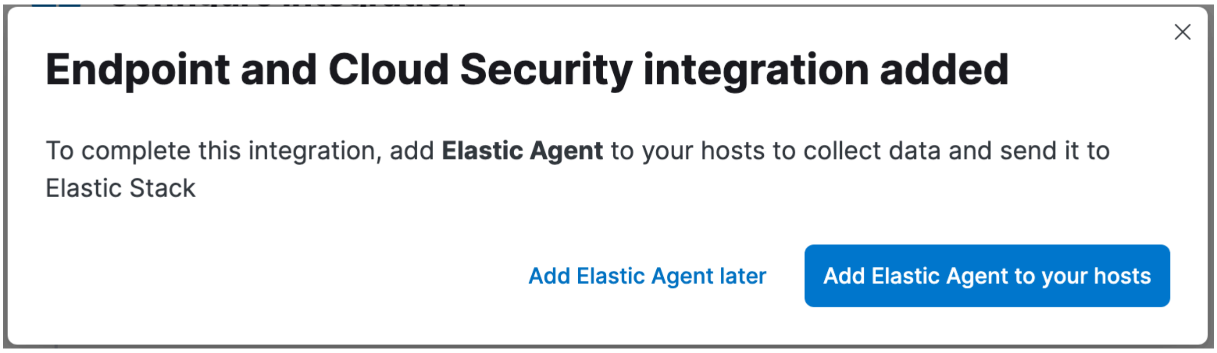
Description: Optional

> Advanced options

We'll save your integration with our recommended defaults. You can change this later by editing the Endpoint and Cloud Security integration within your agent policy.

Figure 13: *Name Integration*

5. You will get a pop-up that asks if you want to add Elastic Agent to your hosts. For now we will avoid this and add agent to hosts at a later step. Click **"Add Elastic Agent later"**.

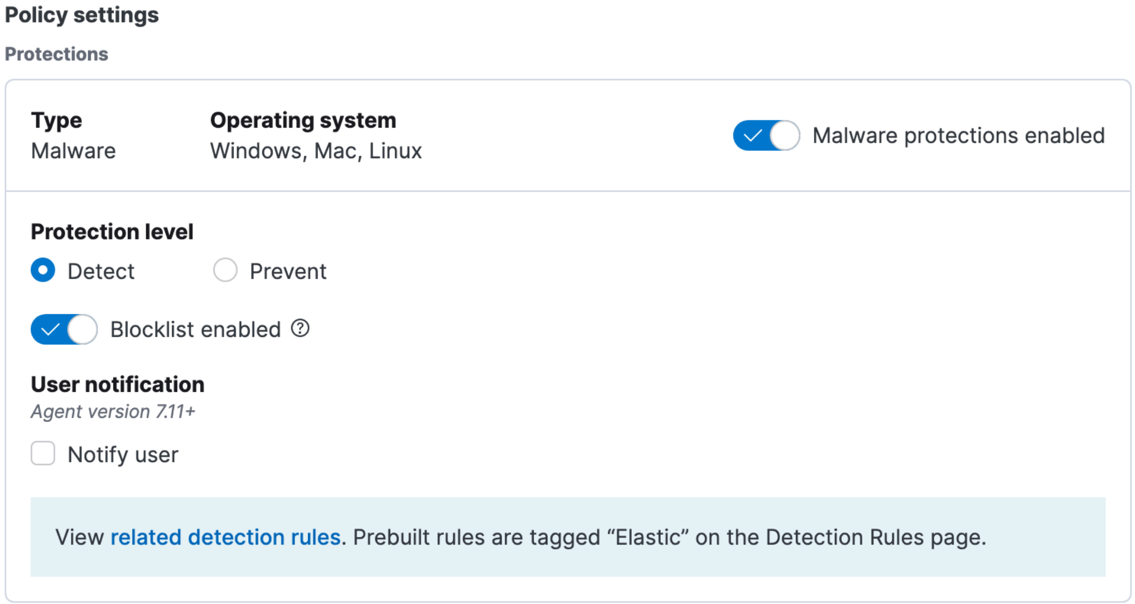


Endpoint and Cloud Security integration added

To complete this integration, add **Elastic Agent** to your hosts to collect data and send it to Elastic Stack

Figure 14: *Add Elastic Agent Later*

6. At this point you should be on the Agent policy page for the policy we created in step 1. Click the name of the integration that was just created to modify the protection settings.
7. Scroll down to the **Policy settings** section and enable/disable the protections that are desired. One the protections that are enabled, set the protection level to **Detect**



Policy settings

Protections

Type	**Operating system**	✓ Malware protections enabled
Malware	Windows, Mac, Linux	

Protection level

● Detect ○ Prevent

✓ Blocklist enabled ⓘ

User notification

Agent version 7.11+

□ Notify user

View related detection rules. Prebuilt rules are tagged "Elastic" on the Detection Rules page.

Figure 15: *Set to Detect*

8. Look through any other settings you would like to modify and when finished click the **"Save integration"** button on the bottom-left of the page.

Step 3: Add the Osquery Manager Integration

1. On the Agent policy page click the blue **"Add integration"** button on the right-hand side.
2. Search for "osquery" in the middle of the page and click the Osquery Manager integration card.
3. On the Osquery Manager integration overview page, click the blue **"Add Osquery Manager"** button on the upper-right side.
4. On the Add Osquery Manager integration page, you may modify the integration name if desired, otherwise click the blue **"Save and continue"** button in the bottom-right of the window.
5. Just like the last step, a pop-up will appear asking to add Elastic Agent to your hosts. Please select **"Add Elastic Agent later"**.

Identify Endgame Hosts that will run Elastic Agent in Parallel

Ideally, a small set of x hosts will be selected to run both the Endgame Sensor and Elastic Agent with the policy we just created. As this initial group of hosts run both sensors in parallel the hosts will need to be monitored to gauge the impact of the additional load. As this group of hosts are evaluated as "stable" the next group of hosts that deploy Elastic Agent can be larger in size. This pattern of deploying to larger and larger groups of hosts is the recommended strategy to ensure stability, test that Agent is detecting the same activities on groups of hosts, and make any needed modifications along the way.

Once the initial group of hosts has been identified, get in touch with their administrators to inform them of the plans to deploy Agent and request their assistance to keep an eye on the hosts and their metrics while the rollout is happening.

Identify Test Users

"Test Users" will need to be identified across teams that can confirm their applications work as expected. This includes items such as scripts, development environments, and application performance metrics such as "logon times". Once these users are identified, make them aware of your deployment schedule and how to report issues if there is a noticeable impact to their environment.

Deploy Agents to a Group of Test Endpoints

Prerequisites You will always need:

- **A Kibana user** with `All` **privileges on Fleet and Integrations**. Since many Integrations assets are shared across spaces, users need the Kibana privileges in all spaces.
- Fleet Server[17] **running in a location accessible to Elastic Agent**. Elastic Agent must have a direct network connection to Fleet Server and Elasticsearch. If you're using our hosted Elasticsearch Service on Elastic Cloud, Fleet Server is already available as part of the Integrations Server. For self-managed deployments, refer to Add a Fleet Server[18].
- **Internet connection for Kibana to download integration packages from the Elastic Package Registry**. Make sure the Kibana server can connect to `https://epr.elastic.co` on port `443`. If your environment has network traffic restrictions, there are ways to work around this requirement. See the section on **Air-gapped environments** above for more information.

If you are using a Fleet Server that uses your organization's certificate, you will also need:

- **A Certificate Authority (CA) certificate to configure Transport Layer Security (TLS) to encrypt traffic**. If your organization already uses the Elastic Stack, you may already have a CA certificate. If you do not have a CA certificate, you can read more about generating one in Configure SSL/TLS for self-managed Fleet Servers[19].

Installation steps

> NOTE: You can install only a single Elastic Agent per host.

Elastic Agent is a single, unified way to add monitoring for logs, metrics, and other types of data to a host. It can also protect hosts from security threats, query data from operating systems, and forward data from

[17] https://www.elastic.co/guide/en/fleet/current/fleet-server.html
[18] https://www.elastic.co/guide/en/fleet/current/add-a-fleet-server.html
[19] https://www.elastic.co/guide/en/fleet/current/secure-connections.html

remote services or hardware. As an Endgame admin, you can install Elastic Agent on your endpoints and give Elastic Security[20] a try.

Please note the following requirements before installing the agent:

- The feature is available only for servers with Internet access, as the server will need to download Elastic Agent binaries from https://artifacts.elastic.co/.
- Endpoints should have access to the Fleet server[21], the component that connects Elastic Agent to Fleet, the web-based UI in Kibana used to centrally manage Elastic Agents and their policies.
- This feature is supported only for endpoints running sensor version 3.64 and later.

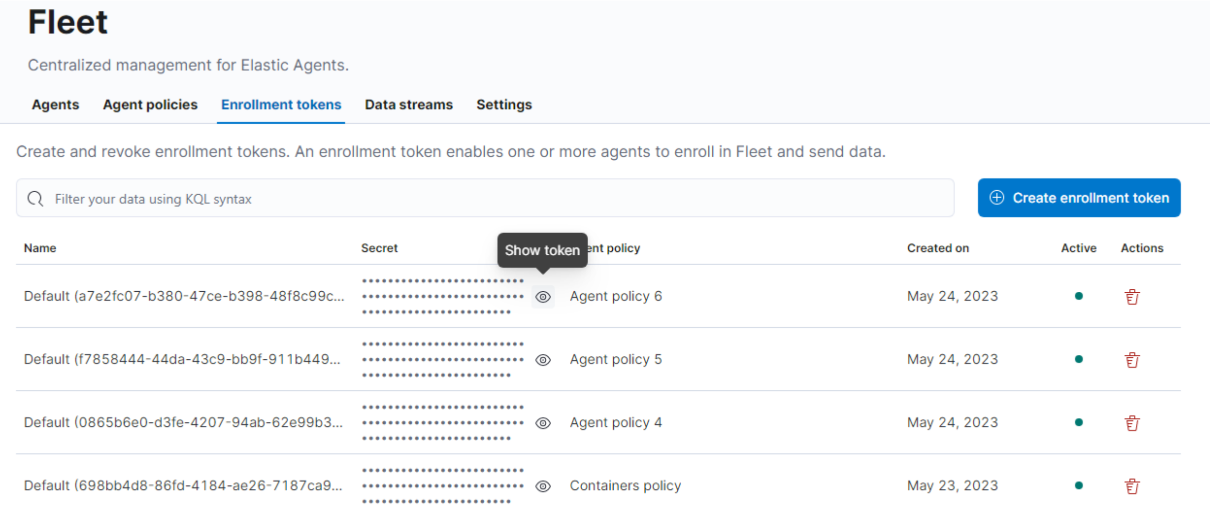
  TIP: Learn more about Fleet, Fleet Server, integrations, policies, and how Elastic Agents are managed in Fleet, check out the documentation here[22].

Using the Endgame SMP to Install Elastic Agent

To install Elastic Agent:

1. Generate a new Fleet token and take note of it. For instructions, refer to Create enrollment tokens.
2. In the Left Navigation toolbar, click the Endpoints button to go to the Endpoint Dashboard.
3. On the Action toolbar, select the appropriate operating system tab to filter the Endpoints list.
4. Select the box to the left of each appropriate endpoint, point to More Actions, then click Respond.
5. Select Install Elastic Agent and enter these advanced configuration details:

- Fleet URL: Enter the Fleet URL.
- Fleet API token: Enter the Fleet API token you noted earlier. To view the list of tokens, go to Fleet > Enrollment tokens.



Create and revoke enrollment tokens. An enrollment token enables one or more agents to enroll in Fleet and send data.

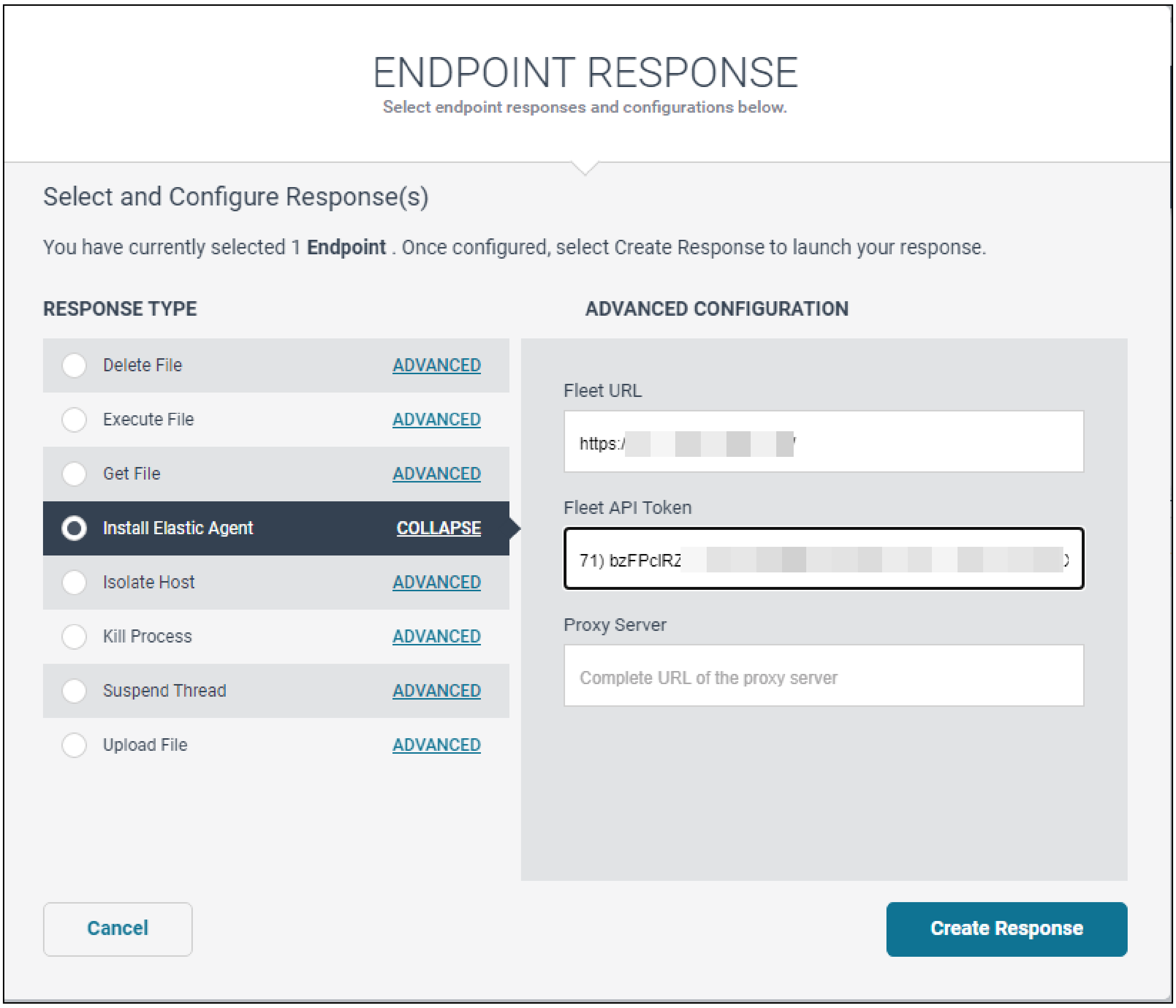
Name	Secret	ent policy	Created on	Active	Actions
Default (a7e2fc07-b380-47ce-b398-48f8c99c...	••••••••••••••••••••••••	Agent policy 6	May 24, 2023	●	🗑
Default (f7858444-44da-43c9-bb9f-911b449...	••••••••••••••••••••••••	Agent policy 5	May 24, 2023	●	🗑
Default (0865b6e0-d3fe-4207-94ab-62e99b3...	••••••••••••••••••••••••	Agent policy 4	May 24, 2023	●	🗑
Default (698bb4d8-86fd-4184-ae26-7187ca9...	••••••••••••••••••••••••	Containers policy	May 23, 2023	●	🗑

Figure 16: *fleet-tokens.png*

[20] https://www.elastic.co/security
[21] https://www.elastic.co/guide/en/fleet/current/fleet-server.html
[22] https://www.elastic.co/guide/en/fleet/current/fleet-overview.html

- Proxy Server (Optional): If there is a proxy between an endpoint and Fleet, enter the complete proxy server URL. If not, leave this field blank.

6. Click Create Response.



ENDPOINT RESPONSE

Select endpoint responses and configurations below.

Select and Configure Response(s)

You have currently selected 1 **Endpoint** . Once configured, select Create Response to launch your response.

RESPONSE TYPE

○ Delete File ADVANCED
○ Execute File ADVANCED
○ Get File ADVANCED
● Install Elastic Agent COLLAPSE
○ Isolate Host ADVANCED
○ Kill Process ADVANCED
○ Suspend Thread ADVANCED
○ Upload File ADVANCED

ADVANCED CONFIGURATION

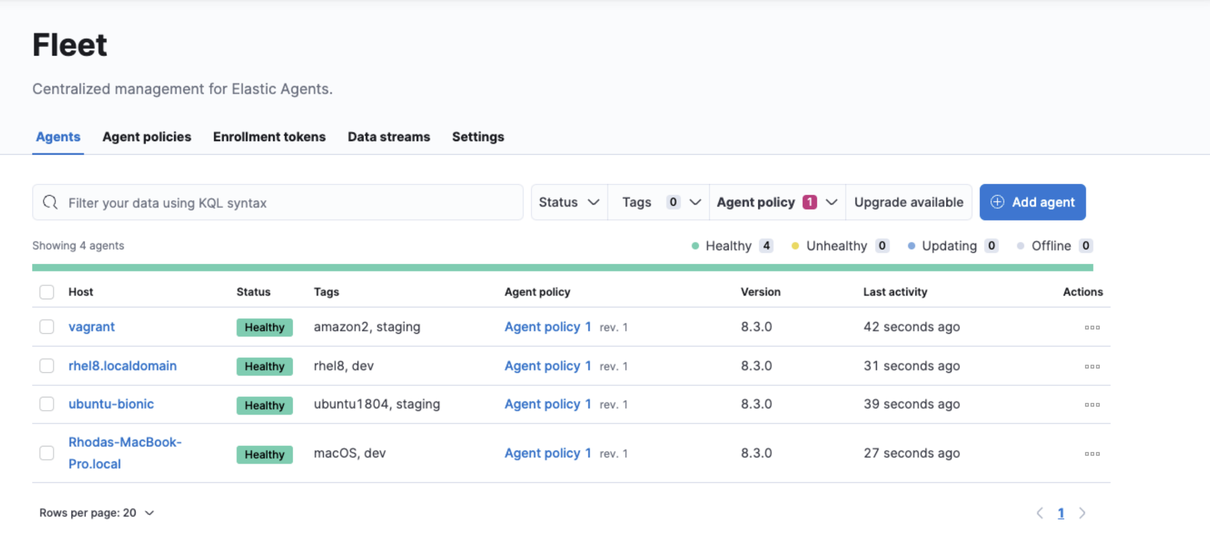
Fleet URL: https:/ [redacted]

Fleet API Token: 71) bzFPclRZ [redacted]

Proxy Server: Complete URL of the proxy server

Figure 17: *install-agent.png*

After the task completes, you should see endpoints being enrolled in Fleet and appearing in the Fleet UI.



Fleet

Centralized management for Elastic Agents.

Agents | Agent policies | Enrollment tokens | Data streams | Settings

Filter your data using KQL syntax | Status | Tags 0 | Agent policy 1 | Upgrade available | Add agent

Showing 4 agents ● Healthy 4 ● Unhealthy 0 ● Updating 0 ● Offline 0

□	Host	Status	Tags	Agent policy	Version	Last activity	Actions
□	vagrant	Healthy	amazon2, staging	Agent policy 1 rev. 1	8.3.0	42 seconds ago	···
□	rhel8.localdomain	Healthy	rhel8, dev	Agent policy 1 rev. 1	8.3.0	31 seconds ago	···
□	ubuntu-bionic	Healthy	ubuntu1804, staging	Agent policy 1 rev. 1	8.3.0	39 seconds ago	···
□	Rhodas-MacBook-Pro.local	Healthy	macOS, dev	Agent policy 1 rev. 1	8.3.0	27 seconds ago	···

Rows per page: 20

< 1 >

Figure 18: *fleet-agents-ui.png*

NOTE: To distribute the load on the endpoints, network, and the Endgame platform, Elastic Agent installation is spread throughout time with a limit on parallel installations. Depending on the number of selected endpoints, it can take up to 72 hours for Elastic Agent to be installed on the endpoint.

Manual Installation of Elastic Agent

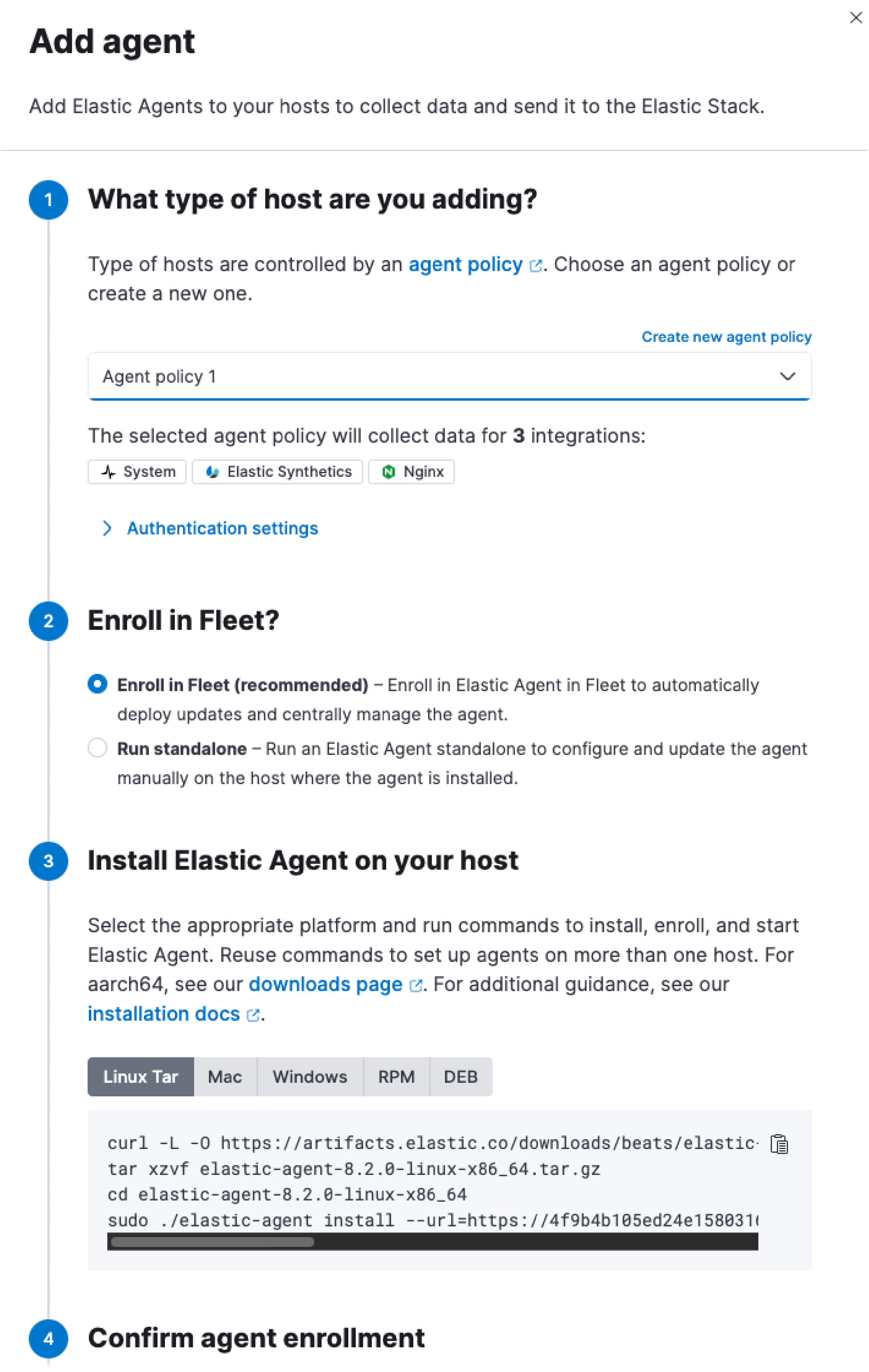
Elastic Agent can monitor the host where it's deployed, and it can collect and forward data from remote services and hardware where direct deployment is not possible.

To install an Elastic Agent and enroll it in Fleet:

1. In Kibana, go to **Fleet** > **Agents**, and click **Add agent**.
2. In the **Add agent** flyout, select an existing agent policy or create a new one. If you create a new policy, Fleet generates a new Fleet enrollment token[23].
3. Make sure **Enroll in Fleet** is selected.
4. Download, install, and enroll the Elastic Agent on your host by selecting your host operating system and following the **Install Elastic Agent on your host** step.

a. If you are enrolling the agent in a Fleet Server that uses your organization's certificate you must add the `--certificate-authorities` option to the command provided in the in-product instructions. If you do not include the certificate, you will see the following error: "x509: certificate signed by unknown authority".

[23] https://www.elastic.co/guide/en/fleet/current/fleet-enrollment-tokens.html



Add agent

Add Elastic Agents to your hosts to collect data and send it to the Elastic Stack.

1 **What type of host are you adding?**

Type of hosts are controlled by an agent policy. Choose an agent policy or create a new one.

Agent policy 1

The selected agent policy will collect data for **3** integrations:

2 **Enroll in Fleet?**

● **Enroll in Fleet (recommended)** – Enroll in Elastic Agent in Fleet to automatically deploy updates and centrally manage the agent.

○ **Run standalone** – Run an Elastic Agent standalone to configure and update the agent manually on the host where the agent is installed.

3 **Install Elastic Agent on your host**

Select the appropriate platform and run commands to install, enroll, and start Elastic Agent. Reuse commands to set up agents on more than one host. For aarch64, see our downloads page. For additional guidance, see our installation docs.

```
curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-
tar xzvf elastic-agent-8.2.0-linux-x86_64.tar.gz
cd elastic-agent-8.2.0-linux-x86_64
sudo ./elastic-agent install --url=https://4f9b4b105ed24e158031
```

4 **Confirm agent enrollment**

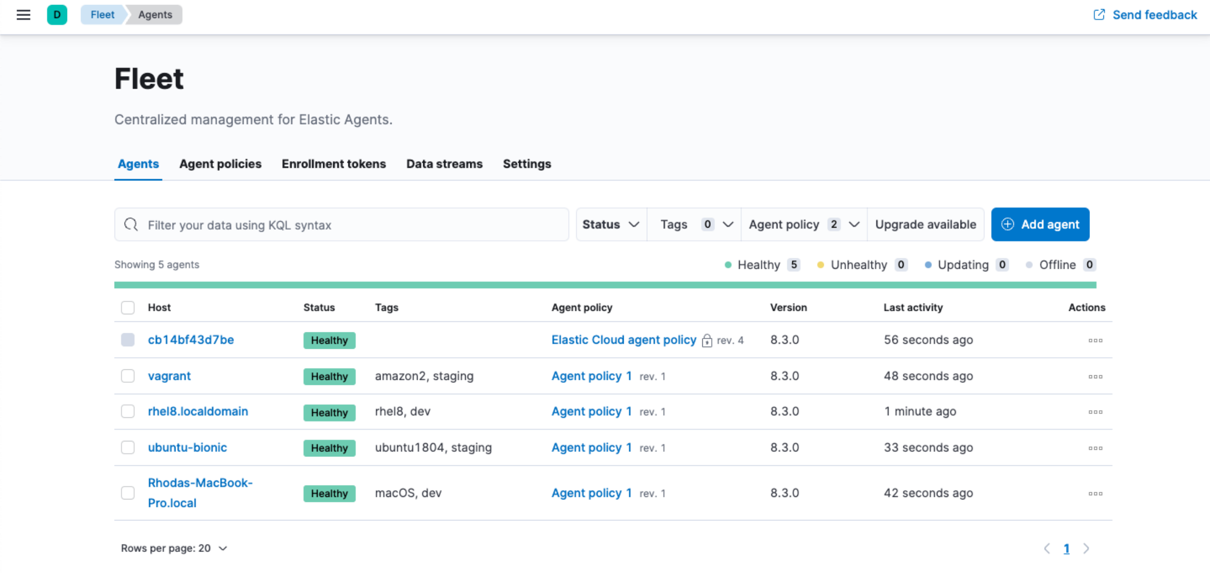
Figure 19: *kibana-agent-flyout.png*

After about a minute, the agent will enroll in Fleet, download the configuration specified in the agent policy, and start collecting data.

Notes:

- If you encounter an "x509: certificate signed by unknown authority" error, you might be trying to enroll in a Fleet Server that uses self-signed certs. To fix this problem in a non-production environment, pass the `--insecure` flag. For more information, refer to the troubleshooting guide[24].
- Optionally, you can use the `--tag` flag to specify a comma-separated list of tags to apply to the enrolled Elastic Agent. For more information, refer to Filter list of Agents by tags[25].
- Refer to Installation layout[26] for the location of installed Elastic Agent files.
- Because Elastic Agent is installed as an auto-starting service, it will restart automatically if the system is rebooted.

To confirm that Elastic Agent is installed and running, go to the **Agents** tab in Fleet.



Fleet

Centralized management for Elastic Agents.

Showing 5 agents

● Healthy 5 ● Unhealthy 0 ● Updating 0 ● Offline 0

	Host	Status	Tags	Agent policy	Version	Last activity	Actions
□	cb14bf43d7be	Healthy		Elastic Cloud agent policy rev. 4	8.3.0	56 seconds ago	···
□	vagrant	Healthy	amazon2, staging	Agent policy 1 rev. 1	8.3.0	48 seconds ago	···
□	rhel8.localdomain	Healthy	rhel8, dev	Agent policy 1 rev. 1	8.3.0	1 minute ago	···
□	ubuntu-bionic	Healthy	ubuntu1804, staging	Agent policy 1 rev. 1	8.3.0	33 seconds ago	···
□	Rhodas-MacBook-Pro.local	Healthy	macOS, dev	Agent policy 1 rev. 1	8.3.0	42 seconds ago	···

Rows per page: 20

Figure 20: *kibana-fleet-agents.png*

TIP: If the status hangs at Enrolling, make sure the `elastic-agent` process is running.

If you run into problems:

- Check the Elastic Agent logs. If you use the default policy, agent logs and metrics are collected automatically unless you change the default settings. For more information, refer to Monitor Elastic Agent in Fleet[27].
- Refer to the troubleshooting guide[28].

[24] https://www.elastic.co/guide/en/fleet/current/fleet-troubleshooting.html#agent-enrollment-certs
[25] https://www.elastic.co/guide/en/fleet/current/filter-agent-list-by-tags.html
[26] https://www.elastic.co/guide/en/fleet/current/installation-layout.html
[27] https://www.elastic.co/guide/en/fleet/current/monitor-elastic-agent.html
[28] https://www.elastic.co/guide/en/fleet/current/fleet-troubleshooting.html

For information about managing Elastic Agent in Fleet, refer to Centrally manage Elastic Agents in Fleet[29].

Monitor Baseline Host Metrics

Elastic Agent running on the same host with Endgame Sensor may consume a noticeable amount of resources depending on the host. It is important to monitor the Elastic Agents on the host to ensure it is not consuming too many resources. Policy configuration, host hardware, and running services are some examples of factors that may affect resource consumption.

Kibana has a built-in monitoring dashboard for Elastic Agent that can be found by navigating to **Dashboard > Search for "Elastic Agent" > click on [Elastic Agent] Agent metrics**.

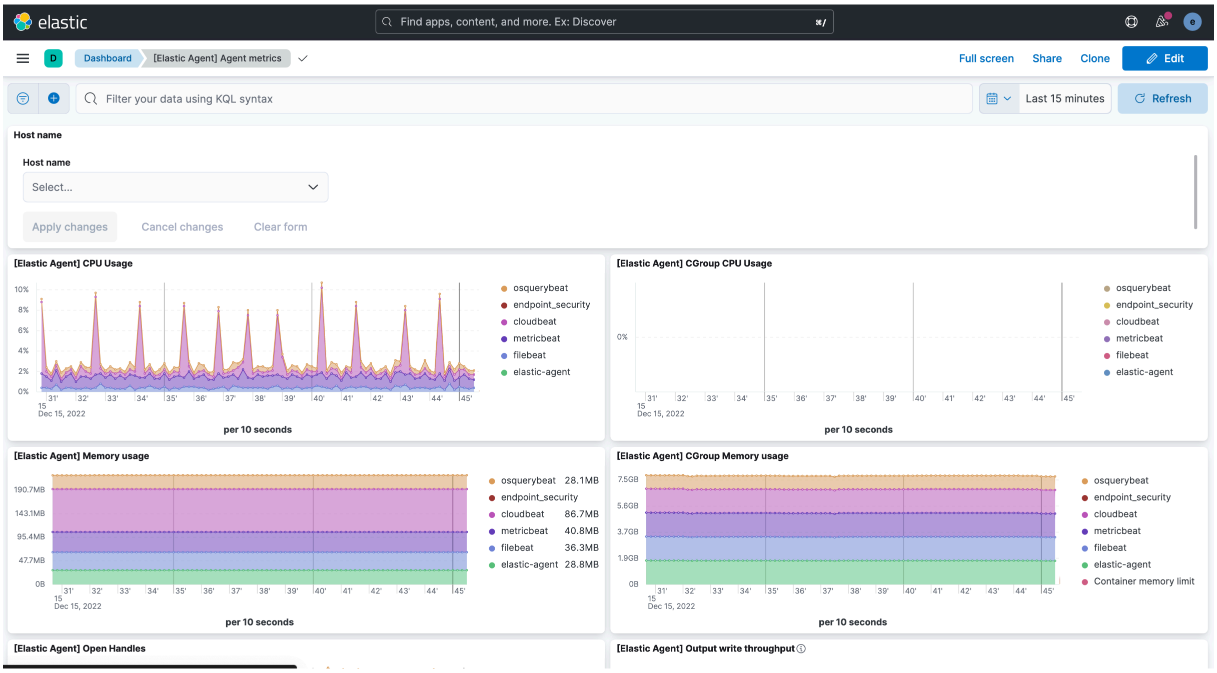


Figure 21: *Metrics Dashboard*

NOTE: There have been some issues with data not showing up in this dashboard due to permissions in the agent. As a workaround, you can use Denfend's own metrics documents to get a CPU histogram and create a visualization. We are working on providing a dashboard for this type of metric along with others in a future release.

These documents can be found in [metrics-endpoint.metrics] datastream

[29] https://www.elastic.co/guide/en/fleet/current/manage-agents-in-fleet.html

Soak period

The "soak period" is really intended to be a window of time to evaluate protection coverage detected by Elastic Agent to ensure the Endpoint/Defend integration is picking up on the same events that Endgame Sensor detects. This can be as long or as short as needed to feel comfortable that the policy settings are tuned to satisfaction.

Once satisfied that coverage with Elastic Agent meets requirements the next batch of hosts should be selected for Agent deployment.

Rinse and Repeat

The process of identifying hosts that will run Agent in parallel, deploying Agent, monitoring and ensuring coverage satisfies requirements needs to be repeated in batches. Ideally these batches can get larger with every repetition, but this is up to preference. Once all desired hosts have Elastic Agent running proceed to the next step.

Moving from Detect to Prevent

At this point we can start the process of removing Endgame. To do so, first we must alter policies in Endgame and Elastic to shift the responsibility of preventing threats.

1. In the Endgame SMP navigate to **Administration** > **Policy**.
2. Go through each policy that is assigned to a sensor and select the **Quick Configure** dropdown button in the upper left-hand corner. Click Enable All (Detect) to turn off preventions.

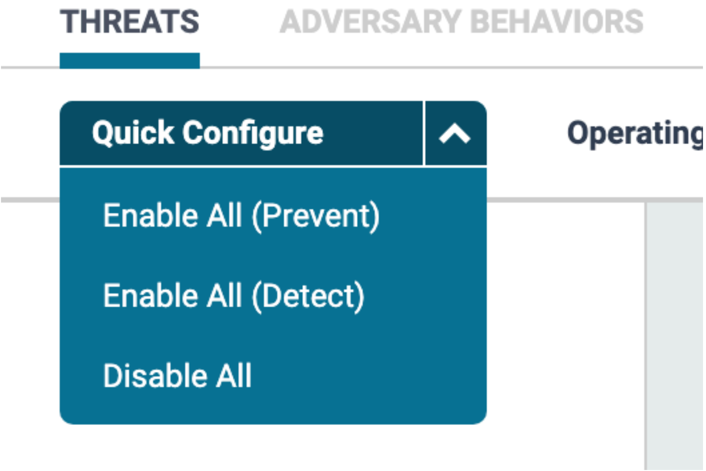
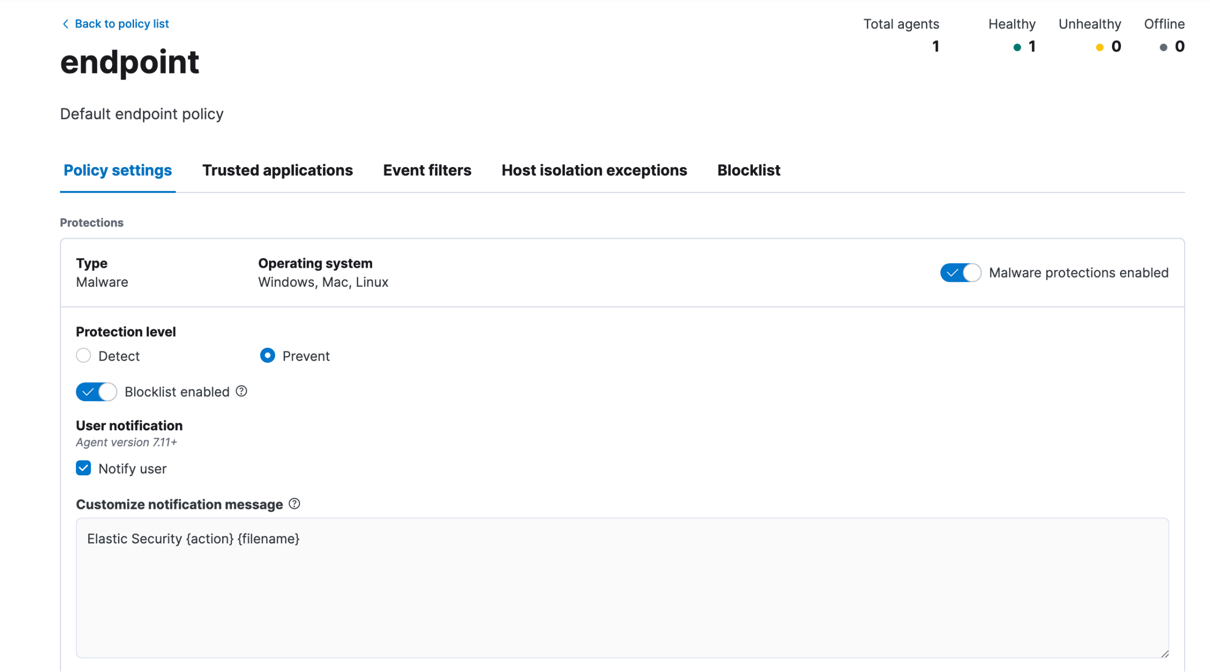


Figure 22: *Endgame Policy Quick Configure*

3. In Kibana navigate to **Security** > **Manage** > **Policies** and select the endpoint policy that is currently running protections in "Detect" mode.
4. Selectively switch the protections desired from "Detect" to "Prevent" and click the blue **Save** button in the bottom right-hand corner.



< Back to policy list

endpoint

Default endpoint policy

Total agents	Healthy	Unhealthy	Offline
1	● 1	● 0	● 0

Protections

Type: Malware

Operating system: Windows, Mac, Linux

✓ Malware protections enabled

Protection level

○ Detect ● Prevent

✓ Blocklist enabled

User notification

Agent version 7.11+

☑ Notify user

Customize notification message

Elastic Security {action} {filename}

Figure 23: *Prevent Policy*

When saved, all Agents will apply the new policy settings and begin stopping threats detected.

Uninstall Endgame

Delete an Endpoint

Deleting an endpoint first uninstalls the sensor from the endpoint, then removes the endpoint from the Endgame platform.

> You can uninstall the sensor but retain the endpoint in the Endgame platform. For more information, see "Uninstall a Sensor." below

To delete an endpoint(s):

1. In the Endpoints list, select the box to the left of each appropriate endpoint.
2. On the Action toolbar, point to More Actions, then click Delete.
3. In the Delete Endpoint(s) dialog box that says, "Are you sure you would like to delete number endpoint(s)? Deleting an endpoint will first uninstall the sensor and then delete all endpoint records from the system (including alerts)." click **Yes**. An "Endpoints successfully deleted" message appears.

4. Click **Finish**.

To delete a single endpoint from the Endpoint Details page, click Take Action, then select Delete Endpoint.

Uninstall a Sensor

Uninstalling a sensor removes it from the endpoint but retains the endpoint in the Endgame platform. You can uninstall a sensor directly via the Endgame platform or an asset management tool, however, it is important to note that if you delete an endpoint, it will first uninstall the sensor from the endpoint, then delete the endpoint from the platform. For more information about deleting an endpoint, see "Delete an Endpoint."

> If the sensor is dissolvable (non-persistent), when you reboot the endpoint, the sensor automatically removes itself.

Uninstall a Sensor via In-band Management In the Endgame platform, you can uninstall a sensor from one or more endpoints simultaneously.

> NOTE: If the sensor is actively communicating with Endgame, it can be uninstalled in-band, even if it was installed out-of-band.

To uninstall a sensor:

1. On the Left Navigation toolbar, click the Endpoints button .
2. In the Endpoints list, select the box to the left of each appropriate endpoint.

> TIP: Ensure the correct operating system tab is selected on the Action toolbar. To select multiple endpoints, choose a bulk selection option from the Current Selection drop-down menu — located directly above the Endpoints list.

4. On the Action toolbar, point to More Actions, then click Uninstall.
5. On the Uninstall Sensors dialog box that says, "Are you sure you would like to uninstall sensors from number endpoints?" click Yes. An "Uninstall request sent" message appears.
6. Click Finish.

> TIP: To verify that the sensor uninstalled successfully, go to the Endpoint Details page, filter the Activity Timeline by ADMIN CONFIGS and verify that an "Uninstall Sensor (Success)" event appears in the activity feed. Or you can select the Unmonitored tab on the Action toolbar to filter the list to endpoints without an installed sensor.

Uninstall a Sensor via Out-of-Band Management It is recommended you only use an out-of-band uninstall method if the sensor is unable to communicate with the platform. If the sensor is actively communicating with Endgame, it can be uninstalled in-band, even if it was installed out-of-band.

Graceful vs. Forceful Uninstall Modes

There are two uninstall modes that can be used to remove the sensor: graceful and forceful. When a graceful uninstall mode is used, the sensor is shut down gracefully. Whether or not the sensor stops, the installer still exits after attempting an uninstall and does not attempt a following installation.

When a forceful uninstall mode is used, the sensor is shut down gracefully, but it is followed by a more aggressive attempt to remove all possible on-disk artifacts. The specific artifacts that need to be removed are based on values from the *.cfg file. As such, an installer file is only able to forcefully remove sensors that were deployed using the same sensor profile the installer file was downloaded from.

> NOTE: If a disguise was used when the sensor was configured, you will need to use the corresponding configuration file to uninstall the sensor.

Uninstall from Windows

To uninstall a sensor from a Windows endpoint:

Locate the previously saved **SensorWindowsInstaller** file from the sensor profile, or download it again. Using your preferred asset management tool, copy the file to the appropriate endpoint(s). Depending on the preferred uninstall mode, run one of the following commands to configure the executable to uninstall the sensor:

True uninstall:

```
SensorWindowsInstaller-<profile name>.exe -c SensorWindowsInstaller-<profile name>.cfg -u true -d false -l
 uninstall.log
```

Force uninstall:

```
SensorWindowsInstaller-<profile name>.exe -c SensorWindowsInstaller-<profile name>.cfg -u force -d false -
l uninstall.log
```

> NOTE: It is recommended to manually type the command instead of copying and pasting.
>
> The -c option uses the specified configuration, and the -u option initiates the true or force uninstall process. While optional, it is recommended to include the -d false option so the installer does not self-delete, and the -l option to create a log file.

Uninstall from Linux

To uninstall a sensor from a Linux endpoint:

1. Locate the previously saved SensorLinuxInstaller file from the sensor profile, or download it again.
2. Using your preferred asset management tool, copy the file to the appropriate endpoint(s).
3. Run the following command to change the modification of the installer: `chmod +x SensorLinuxInstaller -<profile name>`
4. Depending on the preferred uninstall mode, run one of the following commands to configure the executable to uninstall the sensor:

Graceful uninstall:

```
sudo ./SensorLinuxInstaller-<profile name> -c SensorLinuxInstaller-<profile name>.cfg -u true -d false -l
uninstall.log
```

Forceful uninstall:

```
sudo ./SensorLinuxInstaller-<profile name> -c SensorLinuxInstaller-<profile name>.cfg -u force -d false -l
 uninstall.log
```

The -c option uses the specified configuration, and the -u option initiates the true or force uninstall process. While optional, it is recommended to include the -d false option so the installer does not self-delete, and the -l option to create a log file.

Uninstall from macOS

Uninstalling from macOS requires two steps: 1) running the executable to uninstall the sensor and 2) removing the system extension.

To uninstall a sensor from a Mac endpoint:

1. Locate the previously saved SensorMacInstaller file from the sensor profile, or download it again.
2. Using your preferred asset management tool, copy the file to the appropriate endpoint(s).
3. Run the following command to change the modification of the installer: `chmod +x SensorMacOSInstaller -<profile name>`
4. Depending on the preferred uninstall mode, run one of the following commands to configure the executable to uninstall the sensor:

Graceful uninstall:

```
sudo ./SensorMacOSInstaller-<profile name> -c SensorMacOSInstaller-<profile name>.cfg -u true -d false -l
uninstall.log
```

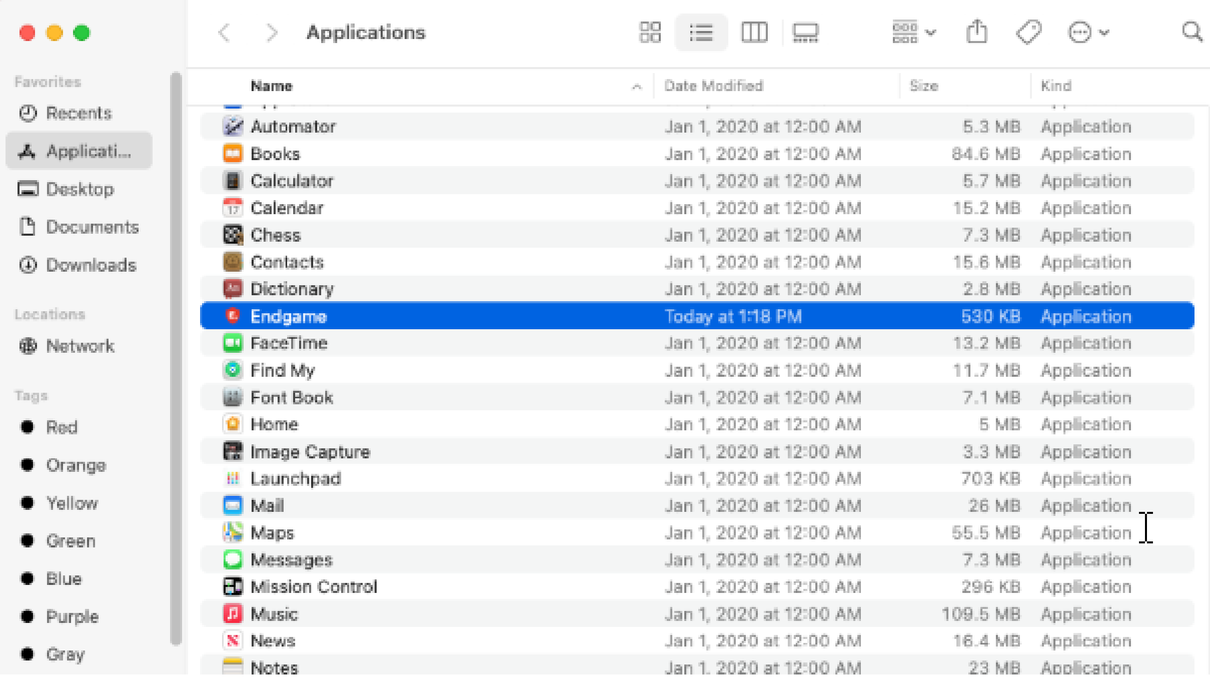
Forceful uninstall:

```
sudo ./SensorMacOSInstaller-<profile name> -c SensorMacOSInstaller-<profile name>.cfg -u force -d false -l
 uninstall.log
```

The -c option uses the specified configuration, and the -u option initiates the true or force uninstall process. While optional, it is recommended to include the -d false option so the installer does not self-delete, and the -l option to create a log file.

To remove the system extension:

1. Open Applications and locate Endgame.



Name	Date Modified	Size	Kind
Automator	Jan 1, 2020 at 12:00 AM	5.3 MB	Application
Books	Jan 1, 2020 at 12:00 AM	84.6 MB	Application
Calculator	Jan 1, 2020 at 12:00 AM	5.7 MB	Application
Calendar	Jan 1, 2020 at 12:00 AM	15.2 MB	Application
Chess	Jan 1, 2020 at 12:00 AM	7.3 MB	Application
Contacts	Jan 1, 2020 at 12:00 AM	15.6 MB	Application
Dictionary	Jan 1, 2020 at 12:00 AM	2.8 MB	Application
Endgame	Today at 1:18 PM	530 KB	Application
FaceTime	Jan 1, 2020 at 12:00 AM	13.2 MB	Application
Find My	Jan 1, 2020 at 12:00 AM	11.7 MB	Application
Font Book	Jan 1, 2020 at 12:00 AM	7.1 MB	Application
Home	Jan 1, 2020 at 12:00 AM	5 MB	Application
Image Capture	Jan 1, 2020 at 12:00 AM	3.3 MB	Application
Launchpad	Jan 1, 2020 at 12:00 AM	703 KB	Application
Mail	Jan 1, 2020 at 12:00 AM	26 MB	Application
Maps	Jan 1, 2020 at 12:00 AM	55.5 MB	Application
Messages	Jan 1, 2020 at 12:00 AM	7.3 MB	Application
Mission Control	Jan 1, 2020 at 12:00 AM	296 KB	Application
Music	Jan 1, 2020 at 12:00 AM	109.5 MB	Application
News	Jan 1, 2020 at 12:00 AM	16.4 MB	Application
Notes	Jan 1, 2020 at 12:00 AM	23 MB	Application

Figure 24: *applications folder.png*

2. Drag and drop the Endgame application to the trash can.

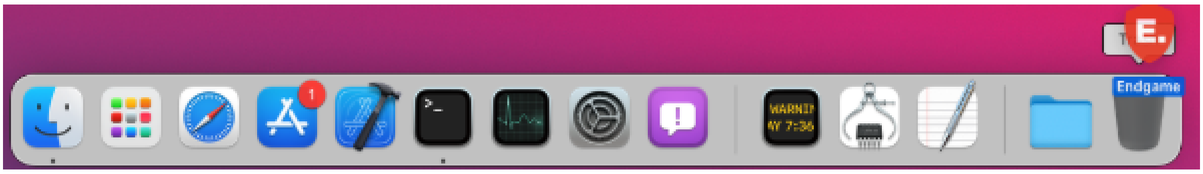
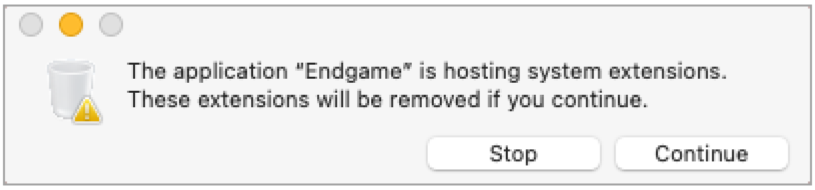


Figure 25: *move to trash.png*

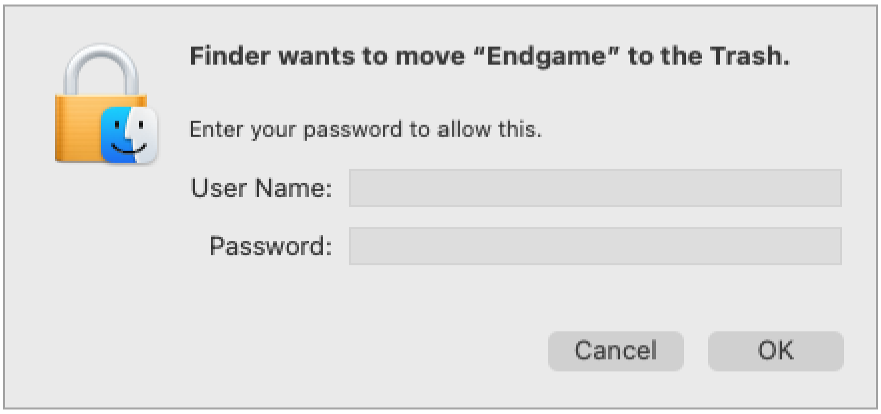
3. In the dialog box that asks to confirm the removal of the system extensions, click Continue.



The application "Endgame" is hosting system extensions. These extensions will be removed if you continue.

Figure 26: *confirm system ext removal.png*

4. Enter your credentials, then click OK to delete and unload the system extension.



Finder wants to move "Endgame" to the Trash.

Enter your password to allow this.

User Name:

Password:

Figure 27: *enter creds to remove.png*

Uninstall from solaris

To uninstall a sensor from a Solaris endpoint:

1. Locate the previously saved SensorSolarisInstaller file from the sensor profile, or download it again.
2. Using your preferred asset management tool, copy the file to the appropriate endpoint(s).
3. Run the following command to change the modification of the installer: `chmod +x SensorSolarisInstaller-<profile name>`
4. Depending on the preferred uninstall mode, run one of the following commands to configure the executable to uninstall the sensor:

Graceful uninstall:

```
sudo ./SensorSolarisInstaller-<profile name> -c SensorSolarisInstaller-<profile name>.cfg -u true -d false
 -l uninstall.log
```

Forceful uninstall:

```
sudo ./SensorSolarisInstaller-<profile name> -c SensorSolarisInstaller-<profile name>.cfg -u force -d
false -l uninstall.log
```

> The -c option uses the specified configuration, and the -u option initiates the true or force uninstall process. While optional, it is recommended to include the -d false option so the installer does not self-delete, and the -l option to create a log file.

Game Over

From here the SMP can be decommissioned. If the SMP is running on-prem, this is as simple as wiping the drive containing the SMP and repurposing the server with a fresh OS install. If running in the Cloud, create an SDH for Endgame, and Elastic SREs will decommission the hosted instance.